Natural Language Processing by Reasoning and Learning

Pei Wang

Temple University, Philadelphia PA 19122, USA
http://www.cis.temple.edu/∼pwang/

Abstract. This paper reports the preliminary experiments in a general-purpose reasoning system to carry out natural language comprehension and production using a reasoning-learning mechanism designed to realize general intelligence.

1 Introduction

This paper describes a new approach for Natural Language Processing (NLP) in a system aimed at the realization of Artificial General Intelligence (AGI).

In the past decades there are two major approaches in NLP:

- The symbolic approach, which treats a natural language as a formal language defined by a formal grammar [1].
- The statistical approach, which treats a natural language as a stochastic process governed by hidden probabilistic distributions [2].

Though both approaches have made progress and produce valuable theoretical and practical results, they are still far away from their goal. A major problem in them is to treat language in isolation, without considering much of its relations with other cognitive processes, such as reasoning, learning, categorization, etc.

The new approach to be introduced in this paper is based on two recent movements in the related fields: cognitive linguistics and AGI.

Cognitive linguistics differs from the traditional (computational or statistical) linguistics in its following hypotheses [3]:

- language is not an autonomous cognitive faculty,
- grammar is conceptualization,
- knowledge of language emerge from language use.

AGI differs from the mainstream AI in its stress on the general-purpose nature of intelligence, as well as the holistic or integrative approaches to achieve intelligence [4, 5].

In the following, I will explain how to introduce NLP capability into NARS, which is an AGI project developed in the framework of a reasoning system [6, 7]. NARS is based on the theory that intelligence is the ability for a system to adapt to its environment while working with insufficient knowledge and resources. It

means that the system depends on finite capacity in information processing, works in real time, and is open to novel tasks and knowledge. Because of this assumption, NARS is very different from the traditional reasoning systems in several major design issues. Given the length restriction, it is impossible to describe all aspects of NARS in this paper. In the following, only the parts of the system that are directly related to NLP are described. For the other aspects of the project, see [6, 7] and other publications, many of which are available at the author's website. NARS is an open source project, with online demonstrations.

The basic ideas of NLP in NARS can be summarized as the following:

- To represent linguistic knowledge in the same form as other types of knowledge, with various levels of abstraction, so that lexical and grammatical knowledge are unified, and directly associated with semantic knowledge.
- To obtain linguistic knowledge from multiple forms of inference, including deduction, induction, abduction, and revision, so it can be derived from the system's experience in language usage.
- To use linguistic knowledge selectively in NLP tasks, according to their records in usefulness and relevance to the current context, so as to capture the fluid nature of meaning.
- To carry out NLP tasks as other cognitive tasks, using the same domain-independent reasoning-learning mechanism, so as to integrate NLP into a model of general intelligence.

2 Knowledge representation

NARS uses a formal language for both internal representation and external communication, which is dubbed Narsese. One of its special property is that it is a *term-oriented* language, and belongs to the "term logic" school, rather than the "predicate logic" school [6]. The basic component of Narsese is a *term*, an identifier that names a concept, which is a recognizable entity in the system's (internal and external) experience. An atomic term is just a sequence of characters from an alphabet. A Narsese statement relates a few terms to each other, and its basic form is an *inheritance statement* "$S \rightarrow P$", where 'S' is the subject term, 'P' the predicate term, and '$\rightarrow$' the *inheritance* copula, which is a reflexive and transitive relation between two terms. The statement says that S is a specialization of P, and P is a generalization of S. For example, "A robin is a bird" can be represented as "$robin \rightarrow bird$" in Narsese. A variant of *inheritance* is the *similarity* copula, '$\leftrightarrow$', which is reflexive, transitive, and symmetric. For example, "A boat is just like a ship" can be represented as "$boat \leftrightarrow ship$".

To express complicated content in a term-oriented language, Narsese utilizes compound term of various types, each of which is formed from some component terms by a *term connector* that serves a logical function. They include:

Sets. A term can be a set formed by given instances or properties. For example, $\{Pacific, Atlantic, Indian, Antarctic, Arctic\}$ is an *extensional set* that can be used in statement "$\{Pacific, Atlantic, Indian, Antarctic, Arctic\} \leftrightarrow$

ocean" to enumerate the oceans; $[red, round]$ is an *intensional set* that can be used in "$apple \rightarrow [red, round]$" to say "Apples are red and round".

Intersections and differences. A compound term can be specified using the instances (or properties) of existing terms. For example, $(bird \cap [black])$ is an *extensional intersection* representing "black bird", while $(bird - [black])$ is an *extensional difference* representing "non-black bird".

Products and images. Non-copula relations can be expressed by inheritance statements. For example, "Cats eat fish" can be equivalently represented as inheritance statements "$(cat \times fish) \rightarrow food$", "$cat \rightarrow (food /_{-}, fish)$", and "$fish \rightarrow (food /cat, _{-})$". Here $(cat \times fish)$ is a *product* that represents the relation between "cat" and "fish", $(food /_{-}, fish)$ is an *extensional image* that represents "things that take fish as food", and $(food /cat, _{-})$ is an *extensional image* that represents "things that cats take as food".

Narsese allows a statement to be used as a compound term to form a "higher-order" statement. For instance, "John knows that the Earth is round" can be represented as "$\{John \times (\{Earth\} \rightarrow [round])\} \rightarrow know$", where *know* is a relation between a cognitive system *John* and a statement "$\{Earth\} \rightarrow [round]$". Similar to the treatment in propositional logic, compound statements can be composed from simpler statements, using statement connectors *conjunction* ($\wedge$), *disjunction* ($\vee$), and *negation* ($\neg$). Furthermore, two "higher-order" copulas are defined between statements: the *implication* copula, '$\Rightarrow$', intuitively stands for "if-then", and the *equivalence* copula, '$\Leftrightarrow$', for "if-and-only-if", though they are not defined exactly as in propositional logic [7].

Furthermore, a Narsese term can also represent a *variable* that can be instantiated by another term, an *event* that is a relation with a temporal duration, and an *operation* that can be executed by the system itself. The definitions and treatments of these terms are described in [7], and some of the ideas come from logic programming [8].

Since NARS is based on the assumption of insufficient knowledge and resources, in it the semantic notions "meaning" and "truth-value" are specified according to an "experience-grounded semantics" [6, 7]. The meaning of a term is defined by its role in the system's experience. The truth-value of a statement is the evidential support the statement gets from the experience, and is written as $\langle f, c \rangle$, with two factors: a *frequency* value in [0, 1] indicating the proportion of positive evidence among all available evidence at the current time, and a *confidence* value in (0, 1) indicating the proportion of current evidence among all evidence available at a "time horizon", after the coming of a constant amount of future evidence.

3 Inference rules

As a term logic, NARS depends on *syllogistic* rules. Such a rule takes two premises (which contain a common term) to derive a conclusion (which is between the other two terms). When the copula involved is *inheritance*, there are

three basic syllogistic inference rules:

	deduction	**abduction**	**induction**
first premise	$M \to P$	$P \to M$	$M \to P$
second premise	$S \to M$	$S \to M$	$M \to S$
conclusion	$S \to P$	$S \to P$	$S \to P$

These rules are named using the terminology introduced by Peirce [9], though the exact definitions of the rules in NARS are not the same as his. In NARS, the *deduction* rule extends the transitivity of the inheritance copula from binary to many-valued, while the *induction* rule and the *abduction* rule can be seen as "reversed deduction", obtained from the *deduction* rule by exchanging the conclusion with a premise, then renaming the terms.

Each inference rule has an associated truth-function to calculate the truth-value of the conclusion from the truth-values of the premises. For the current discussion, it is enough to know that the *deduction* rule is *strong*, as the confidence value of its conclusion is relatively high (it can approach 1), while the other two rules are *weak*, as they only produce conclusions with low confidence values (less than 0.5). If the truth-values are omitted and the rule is applied among binary statements, the strong rules are still valid, but not the weak rules.

When the conclusions about the same statement are derived from different evidential basis, NARS uses a *revision* rule to calculate the amount of the accumulated evidence and the corresponding truth-value. This mechanism allows the weak beliefs to become stronger, as well as balances conflicting evidence. NARS can handle inconsistent beliefs, and that is a major advantage it has over approaches based on probabilistic approach [10], since to maintain the consistency (either in the logical sense or in the probabilistic sense) among the beliefs of an AGI system is not feasible (due to its resource demand), no matter how much it is desired. The consideration on scalability is a major reason for NARS to use *local* inference rules to handle uncertainty, rather than the *global* rules required by probability theory, such as Bayesian conditioning.

When the *inheritance* copula '$\to$' in the above rules is replaced by the *implication* copula '$\Rightarrow$', the inference rules remain valid in NARS. This group of rules is isomorphic to the previous group, in the sense that the truth-functions are the same, though the meaning of the sentences is different, due to the use of different copulas. When one term is taken to mean "something that the system knows" and is implicitly represented, a third group of rules can be obtained:

	deduction	**abduction**	**induction**
first premise	$S \Rightarrow P$	$P \Rightarrow S$	P
second premise	S	S	S
conclusion	P	P	$S \Rightarrow P$

This last group is closer to how these three types of inference are specified in the current AI research, in the framework of propositional logic [11, 12]. The above syllogistic rules show a common principle: in its conclusion, each rule

summarizes the evidence provided by the premises, and the conclusion usually contains compound terms that are not in the premises.

Beside the above rules, there are other inference rules in NARS for the other copulas or term connectors, as well as rules for variable treatment (unification, introduction, and elimination), plus temporal and procedural inference [6, 7].

4 Memory and control

In NARS, a *concept* C_t is an object uniquely named by a term t and contains all the available sentences that have t as a component. For example, sentence "$robin \rightarrow bird$" is stored and processed in concepts C_{robin} and C_{bird}.

A Narseses sentence expresses a conceptual relation. There are two types of sentences in the system: a *belief* is a statement with a truth-value, which summarizes the system's experience on a conceptual relation; a *task* is a sentence to be processed, which can be a piece of new knowledge to be absorbed, a question to be answered, or a goal to be achieved by executing some operations.

The meaning of a concept is defined by its experienced relationship with other concepts. For example, the meaning of C_{bird} at a certain moment is determined by what the system knows and is thinking about the term *bird*. Restricted by insufficient resources, NARS cannot take all existing beliefs into account when processing every task, but has to use them *selectively*. Therefore, the *current meaning* of a concept is determined by a *subset* of the tasks and beliefs existing in the system that form the *full meaning* of the concept.

NARS can obtain new concepts either from its experience or from its reasoning activity. The initial meaning of a concept is usually simple, though it may become complicated as the system gets more related experience. In particular, the meaning of a composed concept may gradually become different from its initial meaning, which is directly formed from the meaning of its components.

As a reasoning system, the running of NARS consists of repeated working cycles. In each cycle, the system mainly works on a selected concept. With two statements containing that term as premises, the applicable rules are invoked to derive one or multiple conclusions. Derived conclusions are added into the memory as new tasks. When a new task corresponds to an existing concept or belief, it will be merged into it, so as to contribute to its meaning and/or truth-value, otherwise it will create a novel concept or belief in the system. In general, all compound terms (including statements) defined in Narsese can be generated by the system itself, usually in more than one way. Therefore, even if there is no input during this cycle, the system's reasoning activity still changes the content of the memory, as well as change the meaning of some terms and the truth-value of some statements.

Working in an experience-driven manner, the inference steps to be carried out in every moment is fully determined by the two selected premises in the selected concept. Since NARS is assumed to have insufficient processing time, it cannot fully process every task. Instead, the system dynamically allocates its resources among the tasks, according to their relative urgency and importance

to the system. While processing a task, the system cannot consider all related beliefs, neither. Similarly, it gives the more useful and relevant beliefs higher chance to be applied.

To implement the above control strategy, the system maintains a priority distribution among the concepts in its memory, as well as among the tasks and beliefs in each concept. In each cycle, a concept is selected *probabilistically* according to the priority distribution among the competitors, and then a task and a belief are selected in the concept in the same way. The priority distributions are adjusted according to the immediate feedback collected after each inference step, so as to achieve a high overall efficiency in task processing.

In general, the priority of a data item (a concept, a task, or a belief) depends on three major factors:

its intrinsic quality, such as the clarity and simplicity of a concept, or the confidence of a judgment in a task or belief;
its performance history, that is, whether the item has been useful to the system in the past;
its immediate relevance, that is, whether it is directly related to the current situation, as reflected in the existing tasks.

In this article, it is neither possible nor necessary to describe the details of the measurements involved in the above factors. It suffices to know that the system *selectively* uses its beliefs to process its tasks.

Since NARS is designed to depend on a constant amount of storage space, the number of concepts and the number of tasks and beliefs within each concept have upper bounds. When the memory (or a concept) is full, the concept (or task/belief) with the lowest priority value is removed. This policy and the decay that happens to all priority values form a *forgetting* mechanism. The system not only constantly gets new tasks, beliefs, and concepts from the environment and the reasoning process, but also loses some old ones from time to time.

In a "life-cycle" of the system, by default the system's memory starts empty, though for practical applications it may start with preloaded content. During a life-cycle, the content of the memory changes constantly, as the result of new experience and reasoning activity, and the internal states of the system never repeat. Therefore, if a task is given to the system at different moments of a life-cycle, it may be treated more or less differently.

5 NLP as reasoning

NARS has been formally specified and partially implemented [7]. Recently, some preliminary experiments have been carried out to add NLP capability into the system, which is what this paper reports.

The basic idea behind this attempt is that natural language processing uses similar mechanism as other cognitive activities. Concretely, the innate representation and processing capacity is roughly the reasoning-learning-categorization model specified in NARS. On the other hand, the language-specific knowledge

is mostly learned from experience, rather than built into the system. Therefore, what is needed for NARS to do NLP is to feed properly prepared knowledge into the system, then to guide the system to carry out language processing tasks. Where and how to get the linguistic knowledge is a separate topic that will not be discussed here.

Since in NARS every "object of thought" is represented by a term in Narsese and corresponds to a concept in the memory of NARS, the same is true for all linguistic entities, such as words, phrases, and sentences (as well as other things like phonemes, letters, characters, morphemes, etc., which will be addressed in future research, not here).

For example, the English word "cat" will be represented in Narsese by term **cat**, and correspond to a concept $C_{\mathbf{cat}}$ in NARS. Here bold font is used to distinguish them from term *cat* and concept C_{cat}, since the latter are not directly bounded to any natural language. English words are used for terms and concepts in NARS to make the description easily comprehensible, though in principle *cat* and C_{cat} could be renamed $T3721$ and C_{T3721}, respectively, since their meaning are determined completely by their relations with other terms and concepts, rather than by the identifiers they have in NARS.

Like other terms, the meaning of a term like **cat** is determined by its relations with other terms that have been experienced by the system. As far as the current project is concerned, there are two major types of relation:

Syntactic, where the basic form represents the observed compositional relation between word "cat" and sentence "Cats eat fish". In Narsese, an English sentence is represented as a sequence of words, so "Cats eat fish" can be represented as "$\{\mathbf{cat} \times \mathbf{eat} \times \mathbf{fish}\} \rightarrow sentence$", showing the co-occurrence of the words involved and their order. Issues like tense and the singular/plural difference are ignored at this stage of the project.

Semantic, where the basic form is the observed representational relation between the word "cat" and the term *cat*. In Narsese, this relation can be expressed as "$\{\mathbf{cat} \times cat\} \rightarrow represent$" and "$\{\{\mathbf{cat} \times \mathbf{eat} \times \mathbf{fish}\} \times ((cat \times fish) \rightarrow food)\} \rightarrow represent$", where "*represent*" is a relation between a "sign" (or "symbol") and an "internal" term that does not directly appear in the system's (external) experience, but has been denoted by the sign.

Like other types of knowledge in NARS, each belief on a syntactic or semantic relation is true to a degree, as indicated by its (two-dimensional) truth-value.

The major tasks in NLP, *language understanding* and *language generation*, become question-answering tasks in NARS that contain Narsese sentence like "$\{sentence \times ?x\} \rightarrow represent$" and "$\{?x \times term\} \rightarrow represent$", respectively. In the former, the task is to find a term that the given sentence represents; in the latter, the task is to find a sentence that represents the given term. In Narsese, a term with a '?' prefix is a *query variable* to be instantiated.

In the following, a working example processed by NARS is explained, in a simplified form (compared with the actual form produced by NARS). In this example, the only syntactic information involved is word order, though it does not mean that this approach can only process this type of information.

Initially, NARS is given the following statements with the default truth-value:

$$\{\mathbf{cat} \times cat\} \to represent\ \langle 1, 0.9\rangle \quad (1)$$

$$\{\mathbf{fish} \times fish\} \to represent\ \langle 1, 0.9\rangle \quad (2)$$

$$\{\{\mathbf{cat} \times \mathbf{eat} \times \mathbf{fish}\} \times ((cat \times fish) \to food)\} \to represent\ \langle 1, 0.9\rangle \quad (3)$$

From them, the system can uses the *induction* rule to derive generalized knowledge, and in the process variable terms are introduced into the conclusion:

$$(\{\$1 \times \$2\} \to represent) \Rightarrow$$
$$(\{\{\$1 \times \mathbf{eat} \times \mathbf{fish}\} \times ((\$2 \times fish) \to food)\} \to represent)\ \langle 1, 0.45\rangle \quad (4)$$

$$((\{\$1 \times \$2\} \to represent) \wedge (\{\$3 \times \$4\} \to represent)) \Rightarrow$$
$$(\{\{\$1 \times \mathbf{eat} \times \$3\} \times ((\$2 \times \$4) \to food)\} \to represent)\ \langle 1, 0.29\rangle \quad (5)$$

Terms with the '$' prefix are *independent variables*, and each indicates an arbitrary term. The above conclusions will contribute to the meaning of the phrase "eat fish" and the word "eat", respectively. From them and the following input

$$\{\mathbf{dog} \times dog\} \to represent\ \langle 1, 0.9\rangle \quad (6)$$

$$\{\mathbf{meat} \times meat\} \to represent\ \langle 1, 0.9\rangle \quad (7)$$

the system can derive the following conclusions using the *deduction* rule:

$$\{\{\mathbf{dog} \times \mathbf{eat} \times \mathbf{fish}\} \times ((dog \times fish) \to food)\} \to represent\ \langle 1, 0.41\rangle \quad (8)$$

$$\{\{\mathbf{dog} \times \mathbf{eat} \times \mathbf{meat}\} \times ((dog \times meat) \to food)\} \to represent\ \langle 1, 0.26\rangle \quad (9)$$

These conclusions have relatively low confidence, but can still let the system understand and produce novel sentences it never heard before. With the accumulation of evidence, the repeated patterns in the language will produce linguistic knowledge with higher and higher confidence, though all the knowledge remains revisable by new experience.

More details of these examples can be found at the author's website.

6 Comparisons

Given the complexity of NLP, it is clearly impossible to compare the NARS approach with the others in detail. Here only the major points are listed.

The above approach toward NLP has the following major properties:

- All language-specific knowledge are learned from experience, rather than built into the system. The learning process can follow different rules, including *deduction*, *induction*, *abduction*, *analogy*, *revision*, and so on. The system can directly accept syntactic and semantic knowledge from the outside, so as to bypass some internal learning process. Such input knowledge will still be revised according to the system's experience. Compared to the existing approaches of NLP, this approach may be more similar to how a human being learns a natural language, though NARS is not designed as a descriptive model of the human mind.

- A natural language is treated as a conceptual system that changes over time. New words, phrases, and sentences are introduced from time to time, and the existing ones may change their meaning gradually. Even grammatical conventions may change over time. Some of the changes are long-term and irreversible, such as the learning and evolving of word meaning, while some other changes are short-term and temporary, such as the content-dependent interpretations of words. The system has the ability to adapt to such changes that come from the environment, as well as to initiate such changes by using a language creatively.
- Though the distinction of syntax, semantics, and pragmatics still exist, these aspects of a language are not processed separately in the system. Syntactic knowledge relates the words and phrases in a natural language in various ways; semantic knowledge associates the words and phrases to the terms/concepts; pragmatic knowledge links knowledge to goals (not discussed in this paper). However, they are all conceptual relations in NARS.
- Syntactic knowledge has various generality, with concepts from specific (like "cat" and "eat") to general (like "noun" and "verb") at multiple levels.
- The processing of natural languages is unified with other cognitive processes, such as reasoning, learning, and categorizing. The only specialty is the linguistic knowledge, which is specific to each language and is learned.

Compared with the symbolic approach of NLP, the NARS approach is different in the following aspects:

- It does not require a built-in linguistic competence. There are still innate grammar rules, but they belong to Narsese, the system's native language. All the natural languages are learned, including both vocabulary and grammar.
- Every piece of linguistic knowledge is statistical in nature. All "grammar rules" can have exceptions and counterexamples, while still playing important roles. What the system does is not to find the exact "laws" behind a language, but to summarize its experience in the language, and to use the summary in the most promising way, according to its estimation.

Compared with the statistical approach of NLP, the NARS approach is different in the following aspects:

- Though the truth-value in NARS is not binary, but intuitively similar to probability, it is not defined according to probability theory. In particular, the truth-values of beliefs in NARS at a certain moment do not form a consistent probability distribution. Instead, the beliefs may contain explicit or implicit conflicts [6].
- The truth-value calculation is carried out by the inference rules, which are *local* operations in the sense that in each step of inference, only the premises are considered, and system-wide impacts are usually achieved through many inference steps. On the contrary, inference in probability theory are often *global* (such as Bayesian conditioning), which are not affordable for a system working with insufficient knowledge and resources.

– Learning in NARS is incremental and ongoing. As shown by the previous example, the system can learn a new sentence template from a single example, though additional examples can increase the confidence of the conclusion.

Compared with the related works in cognitive linguistics, this new approach has the following properties:

– It realizes the basic ideas in cognitive linguistics in a formal model.
– It provides a unification of NLP and reasoning-learning.

Though there are other inference-based NLP work [11], their grammar and inference rules are very different from those of NARS. Detailed comparisions with them will be left for future publications.

Compared with other AGI projects, this approach does NLP, without using an NLP-specific module. Instead, language processing and other cognitive faculties are coupled at a much deeper level and carried out by a unified mechanism.

Acknowledgment

The author thanks Zack Nolan for comments and suggestions.

References

1. Allen, J.F.: Natural Language Understanding. 2nd edn. Addison-Wesley, Reading, Massachusetts (1994)
2. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA, USA (1999)
3. Croft, W., Cruse, D.A.: Cognitive Linguistics. Cambridge University Press, Cambridge (2004)
4. Goertzel, B., Pennachin, C., eds.: Artificial General Intelligence. Springer, New York (2007)
5. Wang, P., Goertzel, B.: Introduction: Aspects of artificial general intelligence. In Goertzel, B., Wang, P., eds.: Advance of Artificial General Intelligence. IOS Press, Amsterdam (2007) 1–16
6. Wang, P.: Rigid Flexibility: The Logic of Intelligence. Springer, Dordrecht (2006)
7. Wang, P.: Non-Axiomatic Logic: A Model of Intelligent Reasoning. World Scientific, Singapore (2013)
8. Kowalski, R.: Logic for Problem Solving. North Holland, New York (1979)
9. Peirce, C.S.: Collected Papers of Charles Sanders Peirce. Volume 2. Harvard University Press, Cambridge, Massachusetts (1931)
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann Publishers, San Mateo, California (1988)
11. Hobbs, J.R., Stickel, M.E., Appelt, D.E., Martin, P.: Interpretation as abduction. Artificial Intelligence **63**(1-2) (1993) 69–142
12. Flach, P.A., Kakas, A.C.: Abductive and inductive reasoning: background and issues. In Flach, P.A., Kakas, A.C., eds.: Abduction and Induction: Essays on their Relation and Integration. Kluwer Academic Publishers, Dordrecht (2000) 1–27