Natural Language Processing in NARS: Preliminary Experiments

Pei Wang

January 15, 2018

This technical report summarizes my ideas on this topic after the AGI-13 paper [1] and its extended version [2], and also takes the work of Ozkan Kilic [3] into consideration. The major purpose is to specify the next stage of NLP experiments in NARS.

A. Knowledge Representation

Initially, NARS will use **dedicated input/output channels** for texts in each specific natural language, so as to skip the tasks such as speech or character recognition, language identification, and so on. At a future time, NLP can also happen in other (sensory) input channels, where multiple types of input/output are mixed, as well as multiple languages.

The input/output tasks in a NLP channel will initially take the form of sentences in a human language, with the boundary of a sentence explicitly marked. Larger units like paragraph and discourse are implicitly represented. Each sentence consists of a sequence of words, and each of which is a string of symbols from a constant alphabet. At the initial stage, a word is treated as an atom without internal structure, so morphological features like prefix and suffix will be handled at a future stage. Consequently, initially the NLP input is merely a stream of **sentences**, while a sentence is a sequence of **words**. Punctuations are limited to '.', '?', and '!' at the end of a sentence. The same is the case for NLP output. Consequently, a sentence corresponds to a Narsese task.

To represent a sequence in NLP and sensory data, a new Narsese operator **list** ('#') is introduced. Compared to the **product** operator ('*'), there are the following similarities and differences:

- *Product* is a general compound for semantic relations among concepts beyond the built-in copulas, while *list* is reserved for temporal or spatial relations among sensory terms only.

- Both *product* and *list* can take an arbitrary number of components, and recursive structures, as product-in-product and list-in-list, are supported.

- Conceptually a *product* is like a point in a multi-dimensional space where each dimension has its semantic meaning, while in a *list* the order is temporal or spatial (not semantic or conceptual), and the relative location of a component provides no semantic meaning. Therefore, the order of components in a *product* is purely conventional, which is not the case in a *list*.

- *List*, even with a recursive structure, is fundamentally a linear structure that supports 'append' (or 'concatenate'), and allows a variable to be unified with a sub-list. *Product*, like the other compound connectors, is fundamentally a tree structure that doesn't support these operations.

- Both *product* and *list* allow an inverse operation (image) to represent its compositional relation with a components.

There are different types of terms involved in NLP: **linguistic terms** correspond to words and phrases of a specific human language, and **generic terms** are used purely as internal identifier and have no external correspondence. For instance, the word "bird" and the concept *bird* are different terms.

Among the corresponding concepts, there are three types of conceptual relations:

1. **Lexical knowledge** is mainly represented as '**represent**' relations from linguistic terms to generic terms, which is many-to-many and may be handled as a built-in relation with acquired instances;

2. **Syntactic knowledge** is mainly represented among linguistic terms, such as the compositional relation between a list and its components, or adjacency and co-occurrence of terms within the same list;

3. **Empirical knowledge** is mainly represented among generic terms as semantic or compositional relations.

This approach uniformly covers syntax, semantics, and pragmatics. All relevant knowledge is clustered into the corresponding concepts.

A major difference between this approach and the traditional symbolic NLP is whether to give up the dependency on a complete and formal grammar. In NARS, the syntactic concepts (like "noun" and "verb") and knowledge (like "A sentence may consists of a noun phrase followed by a verb phrase") are all optional, that is, such knowledge can be acquired, and they do contribute to the system's NLP performance, but their existence is not a precondition of the system's NLP competence. In particular, syntactic and grammatical knowledge are expressed at difference levels (words, phrases, sentences, as well as categories of them), and associated with the semantic and pragmatic knowledge of the concepts involved. The meaning of a compound linguistic term is semi-compositional, as it is determined both by the system's experience with the term as a whole, as well as with its components.

Another major characteristic of NARS is to explicitly separate the linguistic term/concept and the generic term/concept, and leave the semantic knowledge to the latter. Neural networks usually only learn specific mappings at the word/phrase level, without language-independent conceptual knowledge and context-sensitive pragmatics knowledge, as exemplified by the neural translation systems, where sentences of two languages are directly mapped into each other, without an intermediate semantic representation.

B. Inference Rules，Term Constructors, and Mental Operators

The inference rules of NARS treat the linguistic terms as the other terms.

List has constructors like the 'append' predicate in Prolog, which may have multiple versions for different components.

In the future, special mental operators or additional constructors may be introduced, such as to handle morphological operations on words.

C. Inference Tasks

Most NLP tasks are represented as questions on the *represent* relation, either with a linguistic term to be understood or a generic term to be expressed. In the processing of such a task, all types of knowledge (lexical, syntactic, and empirical) may be accessed or derived.

For an understanding task, the system first attempts to find or derive an answer to the sentence as a whole, and at the same time recursively decompose the sentence into phrases and words, and use the existing *represent* relations to construct a new compound term (statement) as the meaning of the sentence.

In this process, some other concepts and beliefs may be created as by-products. There is no separate parsing and semantic mapping, nor complete parse-tree built. It is possible for multiple answers to be found as a form of ambiguity, and the choice rule can find the best answer when invoked.

NARS has tolerance to ambiguity, as their resolution can wait until a choice is necessary. Even after a choice is made, the other candidate answers will still be in the memory, and will have some effects. This will allow the system to handle puns, metaphors, and the related phenomena. Another form of ambiguity comes from the fluidity of concepts and inter-personal differences, and proper understanding requires estimations about factors like the intention and beliefs of the speaker or source of the sentence.

Learning happens during the understanding process, as the inference activities create new concepts and beliefs, as well as change the meaning of the related concepts by adjusting priority distributions. The construction of meaning is both top-down and bottom-up, with the processing of compounds and components happening in parallel. There is no fixed algorithm to accurately specify the order of the steps involved. Instead, that will be determined at the run time via dynamic resource allocation.

D. Knowledge Acquisition

One approach for NARS to get initial knowledge for NLP is to load WordNet [3].

The knowledge in WordNet is actually a mixture of lexicon, syntactic, and empirical knowledge, as distinguished above. When converted into Narsese, the linguistic terms and generic terms must be explicitly separated. Specifically, a *synset* of WordNet will be represented in NARS by a generic term. A consistent naming convention will be needed for such terms, so generic term can be related to the images of linguistic terms under the *represent* relation.

Initially, only selected types of knowledge in WordNet will be imported into NARS, and truth-values and priority values will all use default values, though “common usage” and “rear usage” should be distinguished using different values wherever possible.

WordNet knowledge has the following major limitations from the view point of AGI that need to be remedied using other knowledge sources:

- There is little context information. Here "context" is not limited to the linguistic materials immediately before and after a word or phrase, but also includes factors like conversational goal, attention allocation, emotional status, background information, external environment, etc. This broad sense of context is the major source of pragmatic knowledge.

- The generic terms generated from WordNet will be heavily bias by English. This result actually shows an important difference between the acquisition of native and foreign languages: the former is learned together with *new* concepts, while the latter with *existing* concepts.

References

[1] Pei Wang, Natural language processing by reasoning and learning, Proceedings of AGI-13, Pages 160-169, Beijing, China, August 2013

[2] Pei Wang, Natural Language Processing by Reasoning and Learning, Technical Report of PAGI #1, January 2014

[3] Ozkan Kilic, Intelligent Reasoning on Natural Language Data: a Non-Axiomatic Reasoning System Approach, Master Thesis, CIS Department, Temple University, May 2015