Naturalness vs. Predictability: A Key Debate in Controlled Languages

Peter Clark, Phil Harrison, William R. Murray, John Thompson

Boeing Research and Technology, PO Box 3707, Seattle, WA 98124, USA
{peter.e.clark,philip.harrison,william.r.murray,john.a.thompson}@boeing.com

Abstract. In this paper we describe two quite different approaches to controlled languages (CLs): A "naturalist" approach, in which CL interpretation is treated as a simpler form of full natural language processing, including ambiguity resolution; and a "formalist" approach, in which the CL interpretation is "deterministic" (context insensitive) and the CL is viewed more as an English-like formal/programming language. Despite the philosophical and practical differences, we suggest that a synthesis can be made in which a deterministic core is embedded in a naturalist CL, and illustrate this with our own controlled language CPL. We conclude that despite the philosophical differences, each approach has substantial benefits to be gained from the other.

1 Introduction: The Naturalist vs. Formalist Debate

There are two quite divergent schools of thought concerning the design of controlled languages. The first, which we call the **"naturalist"** approach, treats CL interpretation as a simpler form of the full natural language (NL) processing task in which ambiguity still resides, only to a lesser extent. One might say that the goal is to make English more tractable (understandable by computers) by simplifying the complexity of the English handled. In this approach, as with full NL, multiple interpretations of a sentence are possible, and the CL interpreter uses all the standard NL machinery to search for a "best" parse and interpretation, but with the task being constrained to just the subset of English covered by the CL. Examples of "naturalist" CLs include our own, called CPL [1] (described later), and (to a degree) CELT [2].

The second approach, which we call the **"formalist"** approach, views a CL more as an English-like formal/programming language that is well-defined, predictable, and easier to use than a normal formal language. One might say that the goal is to make logic more tractable (easier to use by humans) by rendering it in human-readable terms that non-mathematicians can understand. Given that many existing formal languages are somewhat cryptic to untrained users, and good NL processing techniques already exist, it is a natural step to have users work with CLs instead that are more readable and translate deterministically into the formal language. A "formalist" approach would view this as an end in itself, and make no claims that the CL necessarily says anything about NL processing in general. Examples of "formalist" CLs include ACE [3], CLCE [4], and CLIE [5].

Although it might seem that the "naturalist" and "formalist" approaches are two sides of the same coin, the underlying philosophies are quite different. A formalist approach eschews nondeterminism, requiring that the CL translates cleanly and predictably to a formal representation, i.e., there is only one acceptable parse and interpretation for any sentence, there is a single sense for each word (plus part of speech), and interpretation follows a logically defined path. In contrast, a naturalist approach attempts to do NL interpretation "in the small," making many disambiguation decisions heuristically (e.g., prepositional phrase attachment, semantic role labeling), and searching for an overall "best" interpretation. A naturalist might seek to gradually extend the CL with time, taking small steps towards fuller NL understanding, while a formalist might declare the CL complete once sufficient expressivity had been achieved.

These different approaches can produce quite different results. A naturalist CL may be more fluent/natural for the user, but also harder to control because the user cannot always predict the disambiguation decisions the system will make. Conversely, a formalist CL may be easier to control (once the user has learned the disambiguation rules), but may also be somewhat less natural to read and may require the user to understand more about the target representation language and ontology.

At Boeing we have been developing a controlled language called CPL (Computer-Processable Language) which clearly falls in the "naturalist" category. We have also created CPL-Lite, a more constrained version which (unlike CPL) is deterministic and falls into the "formalist" category. We now briefly describe these two languages and discuss the strengths and weaknesses of each. Finally, we suggest a way in which the dichotomy between these two rather different approaches might be resolved, and describe ongoing work at creating such a combination.

2. Two Controlled Language Variants

2.1 CPL - Computer-Processable Language

CPL is a mature and extensive controlled language, used in several projects [1,6]. Briefly, CPL accepts three types of sentences: ground facts, questions, and rules. For ground facts, a basic CPL sentence takes one of three forms:

"**There is|are** *NP*"
"NP verb [NP] [PP]"*
"NP **is|are** *passive-verb [***by** *NP] [PP]*"*

where *verb* can include auxiliaries and particles, and nouns in *NP*s can be modified by other nouns, prepositional phrases, and adjectives. For questions, CPL accepts five forms, the two main ones being "**What is** *NP***?**" and "**Is it true that** *Sentence***?**" For rules, CPL accepts an "**IF** *Sent [***AND** *Sent]** **THEN** *Sent [***AND** *Sent]**" form. Words outside the target ontology's lexicon can be used, in which case CPL uses WordNet and the target ontology's lexicon to find the closest concept to those words. Heuristic rules are used for PP attachment, word sense disambiguation, semantic role labeling, compound noun interpretation, metonymy resolution, and other language processing activities. Output is in a Prolog-like syntax, and the primary target ontology is UT Austin's Component Library [7], although others can be used also.

2.2 CPL-Lite

CPL-Lite was designed for a separate project called Mobius [8] to allow trained knowledge engineers, who understood the target knowledge base, to pose queries in a way that was controllable and also (reasonably) comprehensible to others. While CPL searches for the best interpretation, CPL-Lite is simpler and interpreted deterministically. Each CPL-Lite sentence corresponds to a single binary relation (i.e., slot, predicate) between two entities. Both the grammar and vocabulary are more restricted. For assertions, there are ~140 sentence patterns of three types (below), depending whether the relation appears in language as a noun, verb, or preposition:

"**A|An|The** *slot-noun* **of** *NP* **is** *NP|Adj* "	e.g., "A part of a car is an engine" for has-part(x,y)
"*NP slot-verb NP*"	e.g., "A cell encloses ribosomes" for encloses(x,y)
"*NP* **is** *slot-prep NP*"	e.g., "The block is above the ground" for is-above(x,y)

where:

slot-noun is one of a fixed list of ~100 nouns denoting a relation (e.g., "part")
slot-verb is one of a fixed list of ~20 verbs denoting a relation (e.g., "encloses")
slot-prep is one of a fixed list of ~20 (possibly multiword) prepositions denoting a relation (e.g., "above", "in front of")

Thus these three sentence forms expand to a list of ~140 sentence patterns, 1 per relation, allowing any of the 140 predicates in the underlying ontology to be expressed unambiguously. *NP* is one of ~1000 simple nouns (including a few compound nouns) that map directly to concept in the target ontology, i.e., is in the ontology's associated lexicon. Complex noun phrases are not allowed. In addition the sentence form "*NP verb [NP]*" is allowed, where *verb* is in the ontology's lexicon (mapping to an action/event concept), and with the interpretation that the first and second NP are always the agent and object of the verbal concept respectively, i.e., are interpreted as agent(x,y) and object(x,y). Three question forms are also supported.

3. Comparison, and the Naturalist vs. Formalist Tradeoff

Interestingly, CPL-Lite has the same expressivity as CPL; it is just more verbose and restricted, and requires more knowledge of the vocabulary and structure of the target ontology. It is also more predictable: A knowledgeable user can enter a sentence and know exactly how it will be interpreted. In contrast, CPL is more fluent and tolerant of the user: it uses WordNet to "guess" meanings for unknown words, will use lexical and semantic knowledge to try and perform PP attachment and semantic role labeling correctly, and attempt to resolve metonymy. However, there is also a downside, namely CPL may not interpret the sentence in the way intended by the user, and it may not be obvious to him/her how to reformulate the CPL to correct the error (if indeed the user is aware of the misinterpretation).

As an example of CPL and the corresponding CPL-Lite (for the task of posing physics questions, performed in the AURA application [9]), consider the CPL:

A man drives a car along a road for 1 hour.
The speed of the car is 30 km/h.

CPL interprets "for" to here mean the predicate duration(), "along" to mean path(), and attaches both prepositional phrases in the first sentence to the verb ("drive"). In addition, the target ontology considers speeds as properties of events, not objects, and so CPL interprets the second sentence as metonymy for "The speed of the car's driving is 30 km/h" (i.e., it resolves the metonymy with respect to the target ontology). One can alternatively express the same knowledge in CPL-Lite as follows:

A person drives a vehicle.
The path of the driving is a road.
The duration of the driving is 1 hour.
The speed of the driving is 30 km/h.

Here, the user has referred to "person" rather than "man" as "man" is not in the ontology's lexicon; has explicitly spelled out (the words corresponding to) the target predicates; has removed the PP attachment ambiguity; and has correctly attached the speed to the driving event (rather than the car) as required by the target ontology.

However, while the CPL is perhaps more fluent, we can also illustrate the down-side of a naturalist approach. Consider the (hypothetical) case that the CPL interpreter misinterprets "for" as meaning beneficiary(x,y) in the CPL phrase "for 1 hour.". In this case, CPL's "smarts" have gone wrong, and the user is left either unaware of the error, or aware of it but unsure how to rephrase the sentence to correct the problem. To mitigate this problem, CPL paraphrases back its understanding in CPL-Lite and also as a graph so the user is aware of that understanding, and provides reformulation advice and a library of CPL examples to help the user reword if necessary. In general, "smart" software is a mixed blessing -- it can be very helpful, or frustrating to control, or both (e.g., consider automatic page layout or figure placement software). This is essentially the tradeoff that the naturalist vs. formalist approaches presents.

4. Synthesis: Combining the Two Approaches

While it may seem that the underlying philosophies of the naturalist vs. formalist approaches are incompatible, there is a synthesis which we have made, namely to embed CPL-Lite as a deterministic core within CPL itself. In other words, within CPL, we have included the core set of 140 CPL-Lite sentence patterns described earlier whose interpretation is deterministic (i.e., context insensitive) and easily predicable. Given such a core, the user can work within it or beyond it to the extent that he/she feels comfortable, and can fall back on the core if the "smart" interpretation goes wrong. For example, should the CPL interpreter have misinterpreted "for" in "drives...for 1 hour" as beneficiary(x,y), the user can revert to CPL-Lite to make the desired relation explicit: "The duration of the driving is 1 hour", thus correcting the mistake. In this way, the user both has the flexibility to write in more fluent English, while also being able to resort to a more constrained form when necessary to control the interpretation.

We are currently finishing an evaluation of CPL, as embedded in the AURA system [9], in which the user's task is to take exam-style science questions, re-express them in CPL (including the deterministic CPL-Lite core), and pose them to a knowledge base for answering. While the data is still under analysis, one striking

feature of the data analyzed so far is that although CPL provides for a wider variety of forms than CPL-Lite, users stayed within the CPL-Lite subset in the majority of their sentences (~70% in the data examined so far). The most common examples of going beyond CPL-Lite were using complex noun phrases (e.g., "the direction of the applied force on the object"), using words outside the KB's lexicon (e.g., "car", "horse"), and using metonymy with respect to the KB (e.g., "The speed of the man" for "The speed of the man's movement"). As the users were trained with largely CPL-Lite-style example sentences it is perhaps not surprising that they often stayed within this subset, and did not often venture into more sophisticated language forms, and when they did venture out it was somewhat conservatively. This suggests that for users to feel comfortable going beyond that core, the quality of the interpretation needs to be high, and thus "naturalist" CLs are perhaps mainly appropriate for domain-specific applications where the required level of domain-specific customization can be made.

5. Summary

Although the philosophies underlying the naturalist and formalist approaches differ, there is a strong case to be made that each needs and can benefit from the methods of the other. For a naturalist CL such as CPL, there still needs to be a way for the user to control the interpreter's behavior when he/she knows what target output is needed, and this can be done by embedding a formalist-style core, as we have done by embedding CPL-Lite as a core of CPL. Conversely, for a formalist CL such as CPL-Lite, the CL can sometimes be verbose and disfluent, and require a deeper understanding of the target ontology. Adding a more naturalist-like layer to the CL can alleviate these problems, providing that there is sufficient feedback to the user so that there is no confusion about what the system understood. In fact, in practice there are degrees of non-determinism that might be introduced or rejected (e.g., grammar; word senses; semantic roles), and so in implementational terms there is perhaps more of a continuum between the naturalist and formalist extremes. We thus conclude that although there are two quite different philosophies underlying the design of modern CLs, each has substantial benefits to be gained from the other, and there are good reasons for expanding the dialog between practitioners of each.

Acknowledgements

We are grateful to Vulcan Inc. who supported part of this work through Project Halo.

References

1. P. Clark, P. Harrison, T. Jenkins, J. Thompson, R. Wojcik. Acquiring and Using World Knowledge using a Restricted Subset of English. In FLAIRS'05, 2005.
2. A. Pease, W. Murray. An English to Logic Translator for Ontology-Based Knowledge Representation Languages. NL Processing and Knowledge Engineering, 2003.
3. N. E. Fuchs, U. Schwertel, R. Schwitter, Attempto Controlled English. Proc LOPSTR'98.
4. J. Sowa. Common Logic Controlled English. Technical Report, 2004.
5. V. Tablan et al, User-friendly ontology authoring using a CL. ISWC'08. 2008
6. P. Clark, J. Chaw. Capturing and Answering Questions Posed to a Knowledge-Based System. In: Proc 4th Int Conf on Knowledge Capture (KCap'07), 2007.
7. K. Barker, B. Porter, P. Clark. A Library of Generic Concepts for Composing Knowledge Bases. In Proc 1st Int Conf on Knowledge Capture (K-Cap'01), 2001.
8. K. Barker et al.. "Learning by Reading: A Prototype System". In Proc. AAAI'07, 2007.
9. N. Friedland et al.,. Project Halo: Towards a Digital Aristotle. AI Magazine 25 (4), 2004.