A Call for Executable Linguistics Research [*]

Adam Pease

Articulate Software,
420 College Ave
Angwin, CA 94508, USA
apease [at] articulatesoftware.com

Abstract. This paper mirrors my invited talk at PACLIC-22. It describes a call for a renewed emphasis in work on the logical semantics of languages. It lists some of the computational components needed for symbolic interpretations of language, and of automated reasoning within those semantics. It details existing components that meet those needs and provides short examples of how they might work. It also touches on how open source products can support collaboration, which is needed on a project that has the scope of creating a full semantics of language.

Keywords: ontology, natural language understanding, controlled languages, automated deduction, first order logic

1. Introduction

Talks that state the obvious, or review well-known research are boring and risk losing an audience. Talks that give controversial positions often have the same result. But I'd rather take the dangerous route, in hopes of spurring some new ideas and new research. A further risk is that I'm a computer scientist by training, not a linguist. I may have substantial blind spots in computational linguistics, and there are undoubtedly people I'm not aware of already working in the direction I will advocate, but that provides a big opportunity for me to learn from your feedback on this talk.

I'll focus on a broad goal of language understanding or language processing in Artificial Intelligence. I'll define this as a set of techniques for processing human language that show evidence of the same competencies or behaviors as human language processing. Fundamental to this is the ability to accept statements in language that affect future responses, and the ability to respond to questions that demonstrates prior assimilation of knowledge. I don't believe that a “tabula rasa” approach is feasible in this context; I will take it as a given that a great deal of knowledge must already reside in a practical language understanding system.

There has been considerable research in computational linguistics that takes particular linguistic features and subjects them to semantic analysis, with the goal of specifying a formal semantic interpretation derived from syntactic features. This entire area of research has waned however, in part because it was so difficult to combine these sorts of analyses into a single

[*] While this paper discusses nearly a decade of research, which preludes mentioning all the sponsors and collaborators who have contributed, we hope that we will not slight those not mentioned by listing some of the major supporters of this work, who include, US Army CECOM, Army Research Institute, ARDA and DARPA.

semantic theory. The need for providing some interpretation of all text has moved the computational linguistics field to robust shallow interpretations, rather than brittle and deep ones. A contributing factor has been the need for some very large resources to make it possible to have non-toy implementations of deep linguistic semantic processing. Another issue is that because the scope of linguistic semantics is so large, it's hard to tell if different component theories result in a harmonious total interpretation. We're now at the threshold of being able to address these issues. This is why I call for a new direction of Executable Linguistics Research.

Note that I'm not proposing an exclusive alternative to statistical linguistic methods. There are portions of this general problem that benefit from the marriage of statistical and logical approaches, most notably, word sense disambiguation. I'm proposing a shift in emphasis, recommending that more people concentrate on an approach to linguistics that has been somewhat neglected – a shift to focusing on a more difficult and longer term approach, because the utility of current robust and arguably shallow approaches to understanding are yielding less substantial incremental improvements as time goes on.

I'll first sketch an outline of the products that I think are needed. Then I'll discuss existing resources that can meet some of those needs. Next I'll provide some concrete examples to show how this all might work.

2. What's Needed

A fundamental component of any practical and large scale language understanding system is a large vocabulary. Any system that understands language must be able to identify words. It should know basic relationships among words that form some of the building blocks of meaning – synonyms, antonyms, and which words subsume others' meanings or entail them, at a minimum. Harder to acquire, though no less necessary, is a corpus of groups of words that are “tokens”, which have a single and collective meaning that is more than the sum of their component words, and which repeatedly appear in group form.

For English at least, polysemy is a significant issue in language understanding. While better models and algorithms are certainly needed to handle word sense disambiguation the foundation of most algorithms of this sort is the availability of data on which to train the algorithms. Specifically, there is a need for both balanced and domain specific corpora where words have been manually disambiguated with respect to a lexicon.

All this may be relatively uncontroversial so far. Now for the more controversial part. While word meanings change over time, the vast majority of meanings are constant. While meanings can't be legislated, they can be discovered and described, and will largely remain fixed. Linguists often study language at the margins, where there are changes and differences and scientifically interesting features, but much of language is certainly stable, at least over decades. If our goal is machine understanding, it's not enough to know that one word is more specific than another, we must know in what way it is more specific, and what knowledge logically follows from using the more specific word instead of the more general one.

If we agree that this sort of specific information is needed about word meanings, then the next question is in what form it should be represented. There are broadly two options: statistical representations that specify approximate relationships learned automatically, and logical ones that at least at the moment must largely be crafted by humans. Each general approach has advantages.

Statistical approaches require human effort to create a good algorithm, but then can be run automatically on large data sets without human intervention. Such approaches are robust in terms of coverage, but decidedly not robust in terms of the precision of the data. We may be able to learn entailment relationships one inference deep, but I would venture that truth-preserving inferences from automatically acquired data are a long way away. The combinatorics of inference dictate that even if only 10% of the entailments learned are wrong (and the state of art is more like >50% even for simple entailments (TAC 2008), even a simple five step deduction will usually be wrong.

Logical representations can be truth-preserving, and the consistency of any logical theory can be automatically tested (subject to time limitations of course). The main problem is that the effort to craft theories requires human effort, and that effort is specialized and often expensive. I'll return to the issue of open source development later, but for now, let me just state as a given that the scope of such an effort requires open collaboration with many entities and individuals involved.

One issue with logical representations is that words are not logical terms with mathematically precise definitions (at least in most cases). If we treat words as logical terms, or logical terms as though they were words, we'll have an inaccurate model. I'd suggest that we need both a lexicon and a logical model, and relations between them. It is not enough simply to classify words with a small number of formal terms. Knowing that "water" is a "substance" is not sufficient. Any system for understanding must know the implications, uses and properties of water. It must know that water dissolves some other substances, that people can both swim and drown in it, that it can become ice or steam, and many more facts. So, the logical model must ultimately be as large as a dictionary, and it must be subject to continuous evolution as language changes and expands.

Language does not just consist of isolated words. English has many standard phrases in which the meaning of a phrase is more than the sum of its constituent words. A simple case of this is light or "helper" verbs like "take" as in "take a walk" in which the noun functions to modify the mostly meaningless verb. Note that one cannot simply replace the verb with the verbal form of the noun, since "take a hit" is not the same as "hit". Other examples such as "pay attention" are pairs in which the word choices are specific to a given phrase. We cannot simply make these multi-word lexicon entries since we also have examples like "take a long walk" and "a walk was taken". So we must have a corpus of phrases that have logical templates that are filled in by the remaining context of a given sentence.[1]

Beyond a corpus of words and phrases we must have a way of interpreting the overall semantics of a sentence. The building block of a lexicon, phrase corpus, and logical definitions for each is not sufficient. We must be able to interpret the appearance of subject, object, negation, word morphology, conjunction and disjunction, conditionals, modals and many other features. We must handle a host of possibly more mundane linguistic elements like statements about metric time and numbers with units. There is a wealth of research in this area, but not to my knowledge much effort (with Fuch's ACE (Fuchs, 1999) and Kamp& Reyle's DRS work (Kamp&Reyle,1993) as notable exceptions) in systematizing interpretations of all of these different linguistic elements.

The next challenge once we have a system for converting language into logic is to do something useful with the logic. A key enabler is the availability of the same large logical theory that helped us to define individual words in the first place. Those definitions become the corpus of facts and rules that tie together individual logical assertions, and allow us to deduce new consequences. To cover a useful space of knowledge, this theory must be very large. The larger the theory however, the harder it is for a deductive system to process queries on that theory. We must have deductive theorem provers that use smart and adaptive techniques to order the relevance of knowledge, learn how to segment it, and process queries with great efficiency.

3. What We Have

The preceding discussion is not just an abstract exercise, but a guide to research that exists and research that is needed. For each proposed component we have a solution, but all components would benefit from significant focused, collaborative and open research and development.

[1] Examples taken from (Pease&Fellbaum, in press)

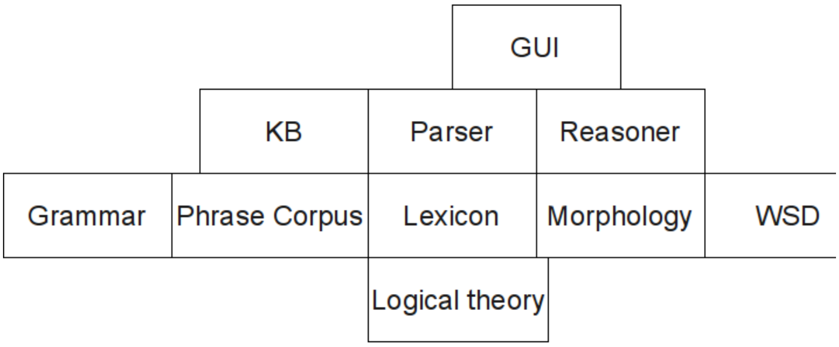

Figure 1: Proposed System Architecture

The foundation of my current work is the Suggested Upper Merged Ontology (Niles&Pease, 2001). It fulfills the need for the logical theory in my proposed model. Although large, it certainly is not large enough. It has some 20,000 terms, but that's much smaller than even a collegiate dictionary. It has 70,000 formal statements (axioms) but that's much smaller than the equivalent of all the definitions in a small dictionary. We use WordNet as our English lexicon (Fellbaum, 1991). SUMO has been mapped by hand (Niles&Pease 2003) to all of WordNet . This has been a massive effort, but much more is needed. There are 10946 mappings from WordNet instances to equivalent SUMO instances and 3774 mappings from synsets to equivalent SUMO classes. The remaining 100459 mappings are from specific WordNet synsets to more general SUMO terms. These mappings are the basis for future work, and are necessary but not sufficient. We need all mappings to be direct equivalences, but this requires a great effort in defining the remaining 100,000 synsets that lack a direct equivalent in SUMO. One might claim that this job is too big to be practical, but efforts of even greater size have happened once people realize a need (wikipedia for example), and join together on community projects. After all, what is the alternative? To persist with shallow understanding based on statistical similarities, and without the ability in computation to deduce logical conclusions from chains of facts as people do? Already, there are efforts such as the one to merge YAGO's 14 million facts (de Melo et al, 2008), which are derived from Wikipedia, into SUMO.

Word sense disambiguation is one of our biggest current challenges. We use WordNet SemCor (Landes et al, 1998), which is a corpus of manually disambiguated sentences from the Brown Corpus (Kucera&Francis, 1967). It is at least two orders of magnitude too small. Many synsets do not appear at all in the corpus, and those that do co-occur with even other common words so few times that there is rarely statistical significance for any given word sense pair. We have also not begun to use any particularly sophisticated methods for employing even the data that we have. There are larger corpora of manual disambiguations, but they are all proprietary. I'll return to that issue later in discussion of open source.

Although we've written in more detail proposing a corpus of phrases (Pease&Fellbaum,in press), the current implementation is a bit ad hoc, and an integral part of the overall English parsing and interpretation system called the Controlled English to Logic Translation system (CELT) (Pease&Murray 2003), (Pease&Li, 2008). Our parser relies on a simple definite clause grammar (Covington 1993) in Prolog augmented with Discourse Representation Theory (DRT) (Kamp & Reyle 1993) to handle anaphor and multiple sentence processing. We use WordNet's “Morphy” algorithm to handle morphology.

CELT takes a certain reductionist approach to handling English grammar. In particular, it does not attempt to handle all of English, as a full semantics of English is simply not possible at the moment. Instead, it is a constructed subset of English. We began with the simplest possible grammar of handling subjects and verbs, then added support for objects and indirect objects, then determiners and quantifiers, conjunction, prepositions etc. At each stage we looked at how we could add a given linguistic feature without creating ambiguity and breaking the understanding of the existing range of grammatical elements. After some five years of development, the scope of what CELT can handle is quite large, although a long way from the full complexity of English. We see CELT as an excellent testbed for theories of linguistic semantics, since any new theory can be tested computationally, and must necessarily interact with a range of theories about other linguistic constructions. As such, it is completely practical.

We use a suite of tools called Sigma (Pease, 2003) to handle processing the logical forms that CELT generates. For the past 6 years we have used the Vampire (Riazanov&Voronkov, 2002) theorem prover. However, as newer provers are now developed that are released open source, we have expanded the set of provers that Sigma includes. Recently, we sponsored a competition on theorem proving performance in SUMO (Pease et al 2008) that inspired the development of a new prover called SInE (Hoder, 2008). Theorem proving performance however remains a significant obstacle to practical implementations of question answering within a logical deductive framework.

4. Diversion: Open Source

While scientific achievement throughout history has often provided the potential for direct financial reward, that potential is great today, and is particular significant in computational linguistics. That profit potential unfortunately leads many researchers and their institutions to control the dissemination of their research, in the hopes of licensing it for profit, or creating a company that develops that research into a product. Profit can be a powerful motivator, but it can also prevent us from engaging in the sort of open collaboration that leads to large research projects and tangible products that engender significant research. Worse yet, the hope of a big financial return leads to much good research remaining unknown and unused, while the researchers also fail to turn the work into a profitable product.

I'll cite one major project I'm aware of in computational linguistics where after many years of US government funding, there is a significant body of work with great potential for reuse. It could result in even more great research, but it remains proprietary. Yet, after several years, the institution has only sold one license for a few thousand dollars. During that same time, they could have collaborated with others on grants, potentially resulting in hundreds of thousands of dollars in new funding, and research results that would have only enhanced the standing of the researchers and their work. This example can be contrasted with the example of WordNet, which having been free from its inception has resulted in near ubiquitous use in English-based computational linguistics, countless funded grants and collaborations for its developers and thousands of publications describing its use on a near unimaginable variety of topics.

5. Example: How it Works

Take the very simple example of “Robert has an orange.” CELT interprets this as

(exists (?orange) **(and** **(attribute Robert-1 Male)** **(instance Robert-1 Human)** **(instance ?orange OrangeFruit)** **(possesses Robert-1 ?orange)))**	o attribute(Robert-1,Male) ^ Human(Robert1) ^ OrangeFruit(o) ^ possesses(Robert-1,o)

Table 1: "Robert has an orange" in SUO-KIF (Pease, 2008) format (left) and conventional logical notation

CELT has a simple database of proper names so it interprets “Robert” correctly. The sense of “orange” as fruit is the most common, and in the absence of a longer sentence or set of sentences, CELT fortunately chooses the a priori most common sense and then retrieves the mapping to the SUMO term of OrangeFruit. Now we ask “Who has a fruit?” (note that the variable that gets the value for “who” is unbound).

(exists (?fruit) **(and** **(instance ?fruit FruitOrVegetable)** **(instance ?who Human)** **(possesses ?who ?fruit)))**	f FruitOrVegetable(f) ^ Human(w) ^ possesses(w,f)

Table 2: "Who has a fruit?" in SUO-KIF format (left) and conventional logical notation

Posing this query to the theorem prover, along with SUMO and the statement asserted above will result in a very simple proof. It relies on the SUMO subclass hierarchy that defines OrangeFruit as a FruitOrVegetable. In this simple example, one could certainly imagine a statistical information retrieval system that uses WordNet's hypernym taxonomy and some simple query relaxation to get the right answer. Pose a more complex query like “What country is between France and Austria” and unless that fact is already explicitly stated, no IR system will find the answer. Of course, for complex queries on large knowledge bases, the logical approach is not guaranteed to find an answer either, but at least it can in theory and there is a clear objective of improving the speed of automated deduction to make reality fulfill the promise.

6. Why Use a Logical Framework

I've explained why this general approach of logical formalization is useful for language understanding software systems. I also believe it's useful for research in language itself. Take for example work on a semantic theory of possession relations in English and the phrases “Robert's nose”, “The car's color”, “Tom's father”. If one is required to state such an interpretation logically, and with recourse to a large theory, some mistakes are easily found and corrected. If we map all possessions to a general relation “owns” and then we ask questions like “What does a car own?” a theorem prover will find automatically that a car owns its color, which is nonsensical. While such mistakes may seem obvious and amenable to human inspection and discovery for an isolated theory of possession, the potential for errors on a much larger theory is much greater. A linguistic theory that is fully implemented with a non-trivial logical theory and lexicon can be tested on a variety of sentences not generated by the creators of the theory.

By collaborating on linguistic research with a common logical theory, linguistics researchers open up the possibility of doing work that directly builds on each others' progress, without the need to create a new harmonization their target representations or notation each time. Working together on a common target semantics enables large-scale executable and testable research in deep linguistic semantics.

Take for example Terence Parsons' excellent book (Parsons, 1990). How do we know that his theories fit with Kamp&Reyle's? While Parsons book has a tighter focus just on event semantics, Kamp&Reyle also cover that area in detail. In neither book is there a formalization (in logic) for relations like subject(x,y) or object(x,y), or the deictic “now”. While these are common notions understood by all linguistics, there are undoubtedly different intuitions at the boundaries. Without a logical theory built on a common logical semantics, we are left to test their compatibility by inspection rather than automation. I cite these books because they are some of the best examples, in my view, of comprehensive linguistic theories with semantics in formal logic. For other work, even more is needed, in my view.

I look forward to hearing from you now about all the good research that I may have missed that meets these goals, and to working with you in the future to help drive linguistics research closer to the ideals that I have described.

References

Covington, M. (1993) Natural Language Processing for Prolog Programmers. Prentice Hall.

de Melo, G., Fabian Suchanek and Adam Pease (2008). Integrating YAGO into the Suggested Upper Merged Ontology. To appear.

Fellbaum, C. (ed.) (1998) WordNet: An Electronic Lexical Database. MIT Press.

Fuchs, N., U. Schwertel, R. Schwitter. (1999). Attempto Controlled English (ACE) Language Manual, Version 3.0, Technical Report 99.03, Department of Computer Science, University of Zurich, August 1999.

Hoder, K., (2008). SinE.0.3. Online description at http://www.cs.miami.edu/~tptp/CASC/J4/SystemDescriptions.html#SInE---0.3

Kamp, H., Reyle, U. (1993). From Discourse to Logic. Kluwer Academic Publishers.

Kucera and Francis, W.N. (1967). Computational Analysis of Present-Day American English. Providence: Brown University Press.

Landes S., Leacock C., and Tengi, R.I. (1998) "Building semantic concordances". In Fellbaum, C. (ed.) (1998) WordNet: An Electronic Lexical Database. Cambridge (Mass.): The MIT Press.

Niles, I & Pease A., (2001). “Towards A Standard Upper Ontology.” In Proceedings of Formal Ontology in Information Systems (FOIS 2001), October 17-19, Ogunquit, Maine, USA, pp 2-9.

Niles, I., and Pease, A. (2003) Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology, Proceedings of the IEEE International Conference on Information and Knowledge Engineering, pp 412- 416.

Parsons, T., (1990). Events in the Semantics of English: A Study in Subatomic Semantics, MIT Press.

Pease, A., and Fellbaum, C., (in press) Formal Ontology as Interlingua: The SUMO and WordNet Linking Project and GlobalWordNet, In: Huang, C. R. and Prevot, L. (Eds.) Ontologies and Lexical Resources. Cambridge: Cambridge University Press.

Pease, A., and Li, J. (2008) Controlled English to Logic Translation. In Theory and Applications of Ontology, ed. Michael Healy, Achilles Kameas, and Roberto Poli, to appear.

Pease, A., and Murray, W., (2003). An English to Logic Translator for Ontology-based Knowledge Representation Languages. In Proceedings of the 2003 IEEE International Conference on Natural Language Processing and Knowledge Engineering, Beijing, China, pp 777-783.

Pease, A., Sutcliffe, G., Siegel, N., and Trac, S., (2008) The Annual SUMO Reasoning Prizes at CASC. Proceedings of IJCAR '08 Workshop on Practical Aspects of Automated Reasoning (PAAR-2008). Volume 373 of the CEUR Workshop Proceedings.

Pease, A., (2003). The Sigma Ontology Development Environment, in Working Notes of the IJCAI-2003 Workshop on Ontology and Distributed Systems. Volume 71 of CEUR Workshop Proceeding series.

Pease, A., (2008). The Standard Upper Ontology Knowledge Interchange Format (SUO-KIF). Available at http://sigmakee.cvs.sourceforge.net/*checkout*/sigmakee/sigma/suo-kif.pdf

TAC (2008). Recognizing Textual Entailment.(RTE) Web site http://www.nist.gov/tac/tracks/2008/rte/

Riazanov A., Voronkov A. (2002). The Design and Implementation of Vampire. AI Communications, 15(2-3), pp. 91—110.