The AAAI 2017 Spring Symposium on
Computational Context: Why It's Important, What It Means, and Can It Be Computed?
Technical Report SS-17-03

Context in Communication

Paul R. Cohen
University of Arizona
Tucson, Arizona

Abstract

The meaning of an utterance, or even a single word, often depends on context, so the inability to represent and process context puts a fundamental limit on how much meaning can be recovered from language. This paper discusses some issues related to context that emerged from the Communicating with Computers program, sponsored by the Defense Advanced Research Projects Agency.

Context has a sunk cost for humans and a prospective cost for machines. Because human evolution has already paid to provide us with context, we have designed or adapted virtually every activity to exploit context and thus recoup its cost. If machines are to perform these activities as well as humans do, then we must expect to pay the prospective cost of providing machines with context. This argument is a slight generalization of one made by Piantadosi, Tily and Gibson (Piantadosi, Tily, and Gibson 2012), who show experimentally that communication is cheaper when context is richer.

If this argument is right – if language evolved to take advantage of context – then many "bugs" in language, which manifest as lexical, syntactic and semantic ambiguity, are actually "features" that exploit context. One shouldn't expect machines to successfully extract the communicative intent of utterances if they can't represent context.

Recognizing the need for new science and technology to give machines access to context, DARPA started the Communicating with Computers (CwC) program. The program defines communication as the process by which an idea in one mind becomes an idea in another. It views language as instructions to construct an approximation of the speaker's idea in the listener's mind. Generally, these instructions are incomplete; missing elements are supplied by context. This is not to say that de-contextualized language is meaningless, only that some meaning is missing.

The sentence "Add another one" is not meaningless, as shown by the sophisticated semantic parse in Figure 1. The TRIPS parser correctly figured out that the sentence is a REQUEST speech act and the agent of the request is *YOU*. It correctly inferred that whatever's to be added is ONE that is differentiated by being OTHER. It chose the INCLUDE sense of the verb "add," though without context it's impossible to say whether this is correct. If the context were doing sums, then "add another one" should invoke an arithmetic sense of "add". Nevertheless, TRIPS got a lot of meaning from "Add another one."

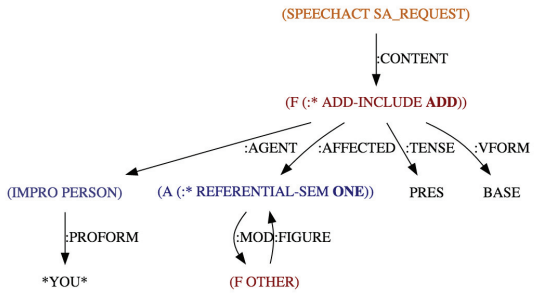

Figure 1: The result of running the TRIPS parser, circa 2016, on the sentence "Add another one." Courtesy: James Allen.

Still, I have long wondered why "deep" semantics of the kind inherent in TRIPS, or the Abstract Meaning Representation, don't seem very semantic to me. The answer is perhaps that only some of the meaning of a sentence can be derived from the sentence itself; the rest comes from context. The parse in Figure 1 is perhaps all the semantics one can hope to get from a de-contextualized sentence. Indeed, recalling the choice of the sense of "add", it might be *more* semantics than is actually warranted. So context adds meaning and also might increase the precision of inferences about meaning.

In context, "Add another one" means more, as illustrated in Figure 2. Suppose someone is building a stack of blocks, and has just placed a block on top of the stack. In this context, "Add another one" probably means, "put another block on the stack." The sense of "add" is PUT, the referent of "one" is the last block placed on the stack, and the referent of "another" is one of the three blocks that are not yet on the stack. But change context just a little, and the meaning changes a lot: Suppose that one comes freshly to the scene in Figure 2 without having observed the stack being built. Now, "Add another one" might mean, "build another stack."

DARPA sponsored the Communicating with Comput-

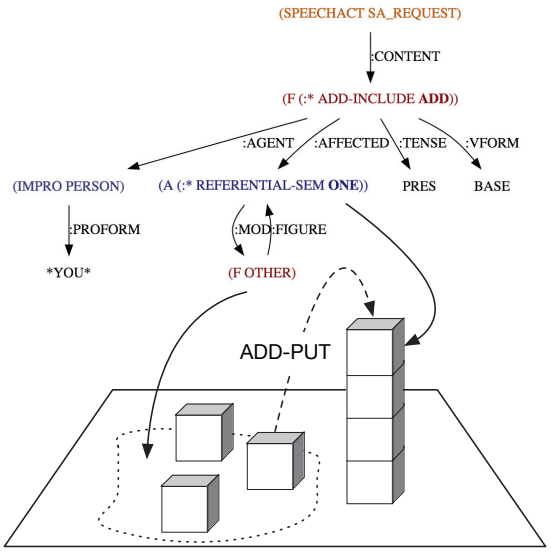

Figure 2: How "Add another one" might be parsed in context.

ers (CwC) program because machines are not yet capable of context-aware language understanding of the kinds illustrated here. CwC works in three broad domains: BlocksWorld involves machines and humans collaboratively building structures from toy blocks. In Bioworld, systems biologists and humans work collaboratively to build and experimentally manipulate simulation models of cell signaling, the molecular processes by which cells function. In ExquisiteCorpse, named for a game invented by Surrealist artists last century, humans and machines take turns contributing to poems, stories, videos and music. In all the CwC domains, gestures and facial expressions are included in utterances.

Lessons about Context from CwC

This section is written as question-answer pairs, where the answer, albeit tentative, is supported by evidence from the CwC program.

Can one measure whether an AI is aware of context, without saying what context *is*? When a capability cannot be objectively assessed (perhaps because it is ill-defined) AI researchers will often invent a task that is thought to depend on the capability and assess performance on this task, instead. So it is with Nasrin Mostafazadeh's Story Cloze Test and ROC corpus of roughly 100,000 five-line stories collected from crowd workers citemostafazadeh-EtAl:2016:N16-1. Each story has a distinct beginning and end, and a causally-related event in between. The contents of the stories are otherwise unconstrained, so the ROC corpus comprises an enormous source of causal common sense. Here is one story with two endings:

> Tom and Sheryl have been together for two years. One day, they went to a carnival together. He won her several stuffed bears, and bought her funnel cakes. When they reached the Ferris wheel, he got down one knee. Tom asked Sheryl to marry him.

Mostafazadeh collected alternative "wrong" endings for several thousand stories; for example, "Tom tied his shoe and left Sheryl" is a semantically odd ending for the story, above. Her Story Cloze Test asks whether machines (and humans) can reliably pick the correct fifth sentence given some or all of the previous sentences as context. Roughly one year ago, the results were stark: Human scores were nearly perfect, machine performance was no better than chance. Today, at least one algorithm can pick the right answer $\approx 70\%$ of the time.

I am glad to see the Story Cloze Test withstand a machine learning assault. If the test is all one hopes for, then superficial representations of context will purchase only small improvements in performance, and big jumps in performance will not be observed until someone develops deeply contentful representations of the context created by the first four sentences.

Should the Cloze Test fall to superficial representations of content, a related test might still prove challenging: Can a machine write a good fifth sentence? Mostafazadeh reports, somewhat discouragingly, that vague fifth sentences (e.g., "Everyone was happy!") have too high a probability of being semantically consistent with context. I think this problem is easily fixed by downgrading fifth sentences that fit multiple contexts/stories, but here, too, superficial hacks might serve (e.g., mention a character, as in,"Tom and Cheryl were happy!")

These concerns notwithstanding, the Story Cloze Test is currently our strongest test of context-aware language understanding, even if we cannot say precisely what context is.

Isn't the context problem just another version of the common sense problem, and so is unlikely to be solved? If by context we mean everything that humans might infer based on voluminous common sense knowledge, then yes, the context problem seems impossible to solve. However, in communication, the speaker judges which parts of the meaning of a sentence to convey in words and which parts the listener could infer given context, and adjusts language accordingly. Moreover, if we think of context as "that which distinguishes one situation from another," then we needn't worry that context is equivalent to all background knowledge.

That said, we are focusing on some chunks of common sense in CwC because they are hard to do without. One chunk is ideas about goals and plans. BlocksWorld and BioWorld are explicitly collaborative problem-solving domains, so it's important that context include some plan-like representations of what the collaborators are trying to do. Language is understood and generated by reference to these representations. The other relevant chunk of common sense is perception and physical action, which we think provides

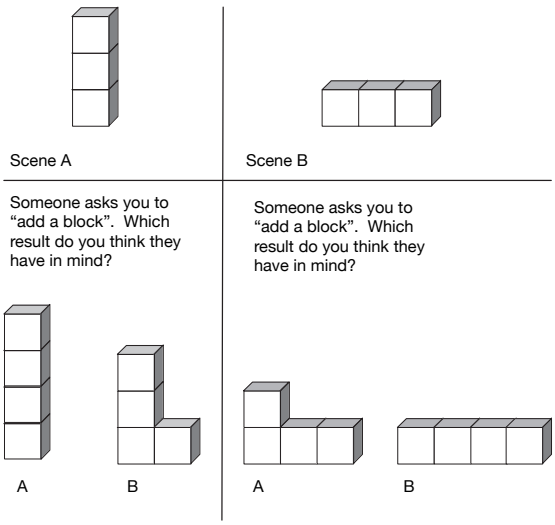

Figure 3: "Add a block" means superficially different things in different contexts.

the semantics for an enormous variety of linguistic constructions, some of which are not physical (e.g., "I am pressing an argument on you").

An early motivation of the CwC program was to have machines capable of representing the context provided by physical scenes. I developed a little quiz, comprising roughly two dozen items like the one in Figure 3. Each has two scenes (A and B) and one sentence (S) with two interpretations (I_A,I_B). The "correct" interpretation of S in A is I_A (and in B it is I_B). For example, most humans who took my quiz think that "add a block" means "put a block on the top of the stack" in scene A and it means "put a block at an end of the row" in scene B. I think this is because scenes A and B are perceived as gestalt arrangements with major and minor axes – a stack and a row – and "add a block" is interpreted as "extend the gestalt by one block along its major axis".

I don't know of any machines that can pass a test of this kind, but CwC participants James Pustejovsky and Nikhil Krishnaswamy have accomplished something similar. They have a semantics for language that depends on encoding physical scenes in terms of their *affordances* (Pustejovsky and Krishnaswamy 2016a; 2016b). In a scene with a cup, a big block and a teaspoon, the sentence "cover the cup" would result in the block, not the spoon, being put on top of the cup, because the system knows what "cover" means and it knows that the spoon, because of its geometry, does not afford covering.

Could we build a "context server"? A context server would be a module that could be queried for context, so that other processes could be "context aware" by querying the server. A context server might store the following kinds of information about the scene in Figure 2.

- Things: There is a stack of four blocks and three individual blocks.
- Relations: The blocks in the stack are related by "on" (or other relations that make the stack a stack); the three other blocks are related by "same size" (or similar relations) but have no apparent spatial organization, though they are a "group" of blocks.
- Focus of attention: The top block in the stack was recently added and is the focus of attention.
- Affordances: Some of the blocks have clear surfaces that afford putting other blocks on them.
- Current and recent intentions, and current and recent actions. For example, the current and recent intention seems to be to build a stack, and the most recent actions each put one block on the stack.

Then, language processing might issue queries to the context server. For example, what are possible references for "another one" in "add another one"? Several things are individuated in the context: The stack, the focus of attention, each of the individual blocks, and the group of blocks. A query to the context server, generated by the phrase "another one," might be something like "return all things of which there is at least one" (because the phrase asks for another). If the phrase were a "a different one" then the query might be something like "return all things that are different (on some features) from the last one." So if the last one were a single green block, the query might find the non-green blocks in the context.

For blocks world applications, Pustejovsky and Nikhil's VoxML language (Pustejovsky and Krishnaswamy 2016b) seems appropriate for encoding context in a symbolic way, but with the rise of neural network methods, it seems quite possible that context will be encoded in subsymbolic ways. To illustrate this issue, let's consider the work of Chloé Kiddon and Yejin Choi on recipes (Kiddon et al. 2015).

The challenge is to have a neural network generate recipes given a recipe title (e.g., baked chicken with preserved lemons) and a list of ingredients, so that all the ingredients are used in the recipe and no non-ingredients are used. An early challenge for Kiddon and Choi was that their system couldn't keep track of which ingredients had been used, so it would generate the same instruction – e.g., "add the chicken" – repeatedly. They built a neural network representation of a checklist of ingredients, and fed it into the network that generates instructions, so as to decrease the probability of instructions that use ingredients that have already been used. It seems perfectly legitimate to call this checklist "context," and it is equally valid to say that the modified language generation system is context-aware. The issue is that, at present, this system is the only one that can use its particular representation of context.

But if we could establish the design of a context server – what it contains, how it is organized, how it is queried, and how it responds to queries – then its *implementation*, in VoxML or neural encoding or other representations, would matter less. We can imagine a neural network-based context server being queried by a more traditional symbolic language system (or vice versa) if we could work out what would be stored in the server and how language, gesture and other communicative moves cause it to be queried.

Conclusion

Context, like meaning, is not well defined, so one must worry that it is a made-up phenomenon, like the boggarts that cause milk to sour and plates to fall off their shelves. Context is blamed for all sorts of maladies, but it isn't necessarily more real than malevolent household spirits. Indeed, there is an imaginary aspect to context: We imagine a boundary between the "primary" materials of intellectual work and the "other stuff" that we call context. In natural language understanding, the primary materials are the words in sentences (and in some cases their lexical semantics) and everything else is context. This distinction seems arbitrary and, as noted earlier, harmful. It is probably reinforced by a fear that context could include anything; that is, context is equivalent to common sense. I think context is more like focus of attention: the content of context is not everything we know but those things that pertain to what we're doing, why we're doing it, which objects and relationships are relevant, and so on. One way to find out how narrow or broad context must be is to build a context server. This suggestion unfortunately perpetuates the perhaps imaginary distinction between primary and contextual material, but it would at least make the boggart real and examinable.

References

Kiddon, C.; Ponnuraj, G. T.; Zettlemoyer, L.; and Choi, Y. 2015. Mise en place: Unsupervised interpretation of instructional recipes. In *EMNLP*, 982–992.

Piantadosi, S. T.; Tily, H.; and Gibson, E. 2012. The communicative function of ambiguity in language. *Cognition* 122(3):280–291.

Pustejovsky, J., and Krishnaswamy, N. 2016a. Generating simulations of motion events from verbal descriptions. *arXiv preprint arXiv:1610.01713*.

Pustejovsky, J., and Krishnaswamy, N. 2016b. Voxml: A visualization modeling language. *arXiv preprint arXiv:1610.01508*.