Extra-Specific Multiword Expressions for Language-Endowed Intelligent Agents

Marjorie McShane and Sergei Nirenburg
Cognitive Science Department
Rensselaer Polytechnic Institute
Troy, NY 12180 USA
{margemc34,zavedomo}@gmail.com

Abstract

Language-endowed intelligent agents benefit from leveraging lexical knowledge falling at different points along a spectrum of compositionality. This means that robust computational lexicons should include not only the compositional expectations of argument-taking words, but also non-compositional collocations (idioms), semi-compositional collocations that might be difficult for an agent to interpret (e.g., standard metaphors), and even collocations that could be compositionally analyzed but are so frequently encountered that recording their meaning increases the efficiency of interpretation. In this paper we argue that yet another type of string-to-meaning mapping can also be useful to intelligent agents: remembered semantic analyses of actual text inputs. These can be viewed as super-specific multi-word expressions whose recorded interpretations mimic a person's memories of knowledge previously learned from language input. These differ from typical annotated corpora in two ways. First, they provide a full, context-sensitive semantic interpretation rather than select features. Second, they are are formulated in the ontologically-grounded metalanguage used in a particular agent environment, meaning that the interpretations contribute to the dynamically evolving cognitive capabilites of agents configured in that environment.

1 Introduction

Language-endowed intelligent agents benefit from access to knowledge of many types of string-to-meaning pairings. The most obvious ones are recorded in the lexicon, which must include not only argument-taking words (along with the lexical, syntactic, and semantic constraints on their arguments) but also a large inventory of multiword expressions (MWEs). MWE is an umbrella term covering many types of entities, a short list of which includes:

- Competely fixed idioms: *It's do or die*
- Idioms with variable slots: [someone] *kicked the bucket*
- Common metaphorical usages that are not semantically opaque: [someone] *is in deep water*
- Frequent phrases that are semantically compositional but for which any other word choice would sound unnatural: *What's for dinner? How can I help you?* (Recording these can speed up analysis as well as ensure the correct paraphrase during generation.)

But what if we were to expand an agent's repository of string-to-meaning pairings even beyond traditional MWEs to full utterances, no matter their linguistic status? What if the agent had access to the correct semantic analyses of a large corpus of inputs such as, "That kid just kicked me in the shins!", "Because I said so!", "If you don't do this within the next five minutes the tank will explode.", and "Scalpel!!"? We hypothesize that memories of successful past language analyses could bootstrap the analysis of new inputs in the ways described in Section 4. We hypothesize further that modeling agents with such a repository is psychologically plausible and, therefore, should be implemented in human-inspired computational cognitive systems.

Proceedings of the Workshop on Grammar and Lexicon: Interactions and Interfaces,
pages 28–37, Osaka, Japan, December 11 2016.

In our earlier writings (e.g., Nirenburg and Raskin 2004; McShane et al. 2005a, 2015) we have described our Ontological Semantics (OS) approach to the lexicon overall, and to MWEs in particular. Some of that material will be summarized here by way of background. But the novel aspect of this contribution involves expanding the notion of useful string-to-meaning pairings to include the agent's repository of previously analyzed inputs. Computing and combining many types of heuristic evidence toward the larger goal of achieving deep semantic analysis contrasts sharply with most current work in NLP, which tends to treat individual phenomena in isolation (MWEs *or* word-sense disambiguation *or* reference resolution) and tends to avoid pursuing the full analysis of text meaning.

The paper is organized as follows. We begin with brief overviews of OS language analysis (Section 2) and the OS lexicon (Section 3). We then consider the text-meaning representations of actual inputs as super-specific MWEs, which can contribute to a knowledge base that supports the analysis of subsequent inputs (Section 4). We conclude with thoughts about how a repository of sentence-to-meaning pairings could serve the wider community as a more fully specified alternative to traditional corpus annotation methods (Section 5). We conclude by commenting on the issues posited to guide the formulation of submissions for this workshop (Section 6).

2 Language Analysis with Ontological Semantics (OS)

The goal of OS text analysis is to automatically generate fully specified, disambiguated, ontologically-grounded text meaning representations (TMRs) of language input. For example, the TMR for the input *John is addressing the situation* is:

```
CONSIDER-1
    AGENT          HUMAN-1
    THEME          STATE-OF-AFFAIRS-1
    TIME           find-anchor-time
    textpointer    addressing
    from-sense     address-v2
HUMAN-1
    HAS-NAME       John
    GENDER         male
    textpointer    John
    from-sense     *proper-name*
STATE-OF-AFFAIRS-1
    textpointer    situation
    from-sense     situation-n1
```

This TMR is headed by a numbered instance of the concept CONSIDER, which is the contextual interpretation of "address". The AGENT of this action is an instance of HUMAN, which is further specified in its own frame as being named John and being male. The THEME of CONSIDER is an instance of the concept STATE-OF-AFFAIRS. The TIME is the time of speech, whose filler is a call to a procedural-semantic routine that attempts to determine when, in absolute terms, the sentence was uttered. If the agent cannot determine that time, then the call to the meaning procedure remains in the TMR, providing an indication of relative time with respect to the other propositions in the text. This is just one example of how OS treats underspecification – an essential aspect of meaning representation. The italicized features are just a couple of the types of metadata stored along with TMRs: the string that gave rise to the frame (textpointer), and the lexical sense used for the analysis (from-sense).

The concepts referred to in TMRs are not merely symbols in an upper-case semantics. They are grounded in a 9,000-concept, property-rich ontology developed to support semantically-oriented NLP, script-based simulation, and overall agent reasoning (McShane and Nirenburg 2012). The information stored about concepts in the ontology is always available to support agent reasoning should that information be needed; however, it is not copied into every TMR. For example, the TMR for the sentence *A dog is barking* will include an instance of DOG (e.g., DOG-1) but it will not include all of the ontological information about typical dogs, such as [HAS-OBJECT-AS-PART: SNOUT, TAIL, FUR], [AGENT-OF GROWL, PLAY-FETCH] , etc. There are three reasons for not copying all of this ontological information into the TMR: first, it is not in the text, and the TMR captures text meaning; second, it is available in the ontology already, should it be needed, making copying redundant; and third, this generic information about dogs might not even apply to this particular dog, which might have no tail, might have never growled a single time in its entire life, and might have no idea why people throw balls into the distance all the time.

A prerequisite for automatically generating TMRs is OS's highly specified lexicon, which we now briefly describe.

3 The OS Lexicon

The OS English lexicon currently contains nearly 30,000 senses. Each sense description contains: metadata for acquirers (definition, example, comments); syntactic and semantic zones (syn-struc and sem-struc) linked by coreferential variables; and, optionally, a meaning-procedures zone that includes calls to procedural semantic routines (for words like *yesterday* and *respectively*).

Consider, for example, the first two verbal senses for *address*, shown in Table 1 using a simplified formalism. Syntactically, both senses expect a subject and a direct object in the active voice, filled by $var1 and $var2, respectively.[1] However, in address-v1, the meaning of the direct object (^$var2) is constrained to a HUMAN (or, less commonly, any ANIMAL), whereas in address-v2 the meaning of the direct object is constrained to an ABSTRACT-OBJECT. The constraints appear in italics because they are virtually available, being accessed from the ontology by the analyzer at runtime. This difference in constraint values permits the analyzer to disambiguate: if the direct object is abstract, as in *He addressed the problem*, then *address* will be analyzed as CONSIDER; by contrast, if the direct object is human, as in *He addressed the audience*, then *address* will be analyzed as SPEECH-ACT.

Table 1. Two verbal senses for the word *address*. The symbol ^ indicates "the meaning of".

address-v1	**address-v2**
anno	anno
definition "to talk to"	definition "to consider, think about"
example "He addressed the crowd."	example "He addressed the problem."
syn-struc	syn-struc
subject $var1	subject $var1
v $var0	v $var0
directobject $var2	directobject $var2
sem-struc	sem-struc
SPEECH-ACT	CONSIDER
AGENT ^$var1 *(sem HUMAN)*	AGENT ^$var1 *(sem HUMAN)*
BENEFICIARY ^$var2 *(sem HUMAN) (relax.-to ANIMAL)*	THEME ^$var2 *(sem ABSTRACT-OBJECT)*

These examples highlight several aspects of the OS lexicon. First, it supports the combined syntactic and semantic analysis of text. Second, the metalanguage for describing its sem-strucs is the same one used in the ontology. And third, the sem-strucs—and, often, the associated syn-strucs—from the lexicon for one language can be ported into the lexicon of another language with minimal modification (McShane et al. 2005a), which greatly enhances the multilingual applicability of the OS suite of resources.

The simplest method of representing lexical meaning in an ontological semantic environment is to map a lexeme directly onto an ontological concept: e.g., dog → DOG. In the case of argument-taking lexemes, the syntactic arguments and semantic roles need to be appropriately associated using variables, as shown our *address* senses above. However, not every word meaning is necessarily represented by a single ontological concept. In some cases, property-based specifications of concepts are provided in the lexicon (for a discussion of what makes it into the ontology, see McShane et al. 2005a). For example, *asphalt* (v.) is described as a COVER event whose THEME must be a ROADWAY-ARTIFACT and whose INSTRUMENT is ASPHALT.

[1] Variables are written, by convention, as $var followed by a distinguishing number. Variables permit us to map textual content from the input to elements of the syn-struc, then link each syn-struc element with its semantic realization in the sem-struc.

```
asphalt-v1
  anno
    definition  "to cover a roadway in asphalt"
    example     "The workers asphalted the country road."
  syn-struc
    subject        $var1
    v              $var0
    directobject   $var2
  sem-struc
    COVER
      AGENT        ^$var1 (sem HUMAN)
      THEME        ^$var2 (sem ROAD-SYSTEM-ARTIFACT)
      INSTRUMENT   ASPHALT
```

Using this lexical sense, among others, to process the input *He asphalted the driveway yesterday* generates the following TMR, presented without metadata:

```
COVER-1
    AGENT        HUMAN-1
    THEME        DRIVEWAY-1
    INSTRUMENT   ASPHALT
    TIME         combine-time (find-anchor-time -1 DAY)  ; find the time of speech and subtract a day
HUMAN-1
    GENDER       male
```

As we see, generating TMRs essentially involves: a) copying the content of sem-strucs into TMR frames; 2) converting bare concept names (COVER) into instances (COVER-1); and 3) replacing variables by their associated concept instances (^$var1 → HUMAN-1).

The lexicon includes a large inventory of MWEs, such as *someone takes someone by surprise.*

```
take-v4
  anno
    definition  "MWE: s.o. takes s.o. by surprise = s.o. surprises s.o. else"[2]
    example     "The clowns took us by surprise."
    comments "Non-agentive subjects are covered by a conversion recorded as a meaning-procedure"
  syn-struc
    subject        $var1
    v              $var0
    directobject   $var2
    pp
       prep     $var3 (root by)
       obj      $var4 (root surprise)
sem-struc
    SURPRISE
      AGENT     ^$var1  (sem ANIMAL) (RELAXABLE-TO ORGANIZATION)
      THEME     ^$var2  (sem ANIMAL) (RELAXABLE-TO ORGANIZATION)
    ^$var3   null-sem+
    ^$var4   null-sem+
meaning-procedure
    change-agent-to-caused-by (value ^$var1)
```

As should be clear by now, the combination of syntactic expectations and semantic constraints renders every argument-taking lexical sense *construction-like*. So, although non-idiomatic argument-taking word senses do not require particular lexemes to be used as their arguments, they do semantically constrain the set of meanings that can be used to fill case-role slots, resulting in what might be thought of as broadly specified constructions. This is not a peculiar side-effect of the theory of OS; instead, we hypothesize that this is how people think about language, and how intelligent agents con-

[2] If the meaning of the subject is non-agentive, the procedural semantic routine *change-agent-to-caused-by* will be triggered. E.g., *His arrival took me by surprise* will be analyzed as SURPRISE (CAUSED-BY COME (AGENT HUMAN) (GENDER male)). An alternative approach would be to simply record another lexical sense that expects a non-agentive subject.

figured to act like people need to learn to think about it. In short, a sufficiently fine-grained lexical specification of argument-taking words – supported by ontological knowledge about the concepts they invoke – is a long way toward being a construction, and constructions are a superclass of what are typically considered multi-word expressions.

Now we turn to the new contribution of this paper: exploring how we can use a repository of stored TMRs as super-specific MWEs that can serve as an additional knowledge base for agent reasoning about language.

4 TMRs as Super-Specific MWEs

When intelligent agents configured within the OntoAgent cognitive architecture carry out natural language understanding, they store the resulting string-to-meaning correlations in a TMR repository. Let us consider the content of the TMR repository alongside the agent's other core knowledge bases, the ontology and the fact repository. The **ontology** describes types of objects and events using the ontological metalanguage; it has no connection to natural language whatsoever. The **TMR repository** contains pairings of text strings with their semantic interpretations, the latter recorded as ontologically-grounded TMRs. Each TMR is supplied with confidence scores along many parameters. These scores permit the agent to reason about whether its level of understanding of the input is sufficient (a) to merit storing the information to memory, and (b) to support subsequent reasoning about action (McShane and Nirenburg 2015). The **fact repository**, for its part, models the agent's memory of concept instances. Like the ontology, it is recorded exclusively using the ontological metalanguage – there are no links to natural language. This is quite natural because agent memories do not derive exclusively from language understanding: e.g., when an agent supplied with a physiological simulation (such as a virtual patient) experiences symptoms, it remembers them as meaning representations with no associated text strings; similarly, when an agent reasons about its task or its interlocutor, it remembers its conclusions and their rationale in the form of meaning representations. In principle, though we are still working out the details, the fact repository should also reflect (a) decision-making about what is worth remembering (i.e., the information should be sufficiently relevant and of sufficiently high quality), (b) the merging of individual memories into generalizations (365 instances of taking a given medication every evening should be merged into the memory of taking the medication every evening for a year), and (c) possibly even forgetting the kinds of things that a regular person would forget – depending on how lifelike one wants the agent to be. In short, the TMR repository is one source of input to the fact repository, and it is the fact repository – along with the ontology – that supports agent reasoning.

It is very difficult for intelligent agents to compute full, completely disambiguated and contextually correct interpretations of natural language utterances – which is presumably the reason why mainstream NLP has sidelined this goal in favor of pursuits with nearer-term payoffs. We will work through just a sampling of the many challenges of full language interpretation that we think will be better handled by agents that are configured to use a TMR repository as a source of evidence.

Challenge 1: Polysemy. Most words are polysemous, with the challenges of disambiguation exploding exponentially with increasing sentence length. But many words are used in frequent combinations. The remembered interpretations of such combinations help human readers save effort in language analysis, and they should serve as the agent's default interpretation as well. For example, when reading *He gave the dog a bone*, any human and human-emulating agent should immediately think "furry canine companion" not "contemptible person" – though the latter interpretation is not excluded based on ontological constraints (one can, in fact, hand a bone to a person).

Stored analyses can be particularly helpful for disambiguation when the input words are used *outside of* a highly predictive dependency structure. For example, disambiguating *horse* in *The horse was eating hay* is straightforward using OS methods since only animate entities can eat things, and the alternative meanings of *horse* (a sawhorse or pommel horse) are inanimate. However, disambiguation is not as easy for *She put some hay down beside the horse*, because "beside the horse" is a free adjunct, and the ontology can be expected to contain only the weakest semantic constraints on where something can be put down. The disambiguation heuristic that people use in such contexts is frequency of co-occurrence.[3] That is, in any context with *hay* – understood as mown, dried grass (not 'bed', as used

[3] This is being explored by distributional semantics; however, since that paradigm works on uninterpreted text strings, it provides no direct support for our agent's goal of ontologically-grounded disambiguation.

in various idiomatic expressions) – any *horse* that is mentioned is likely to be an animal. Our agent can use the TMR repository as a search space, seeking combinations of two or more *words of input* along with their corresponding *concepts in TMR*. The closer the alignment between the incoming and stored input strings – and/or the closer the alignment between candidate interpretations of the incoming string and the interpretation of the stored string – the more confident the result of this method of disambiguation. Formalizing similarity metrics is, of course, key to optimizing this process.

Challenge 2: Non-canonical syntax. The OS lexicon records syntactic expectations for argument-taking words such as verbs. For example, one sense of *manage* expects an infinitival complement (xcomp) and is used to analyze inputs like *He managed to close the door.* But what if an input says *He managed close the door,* which lacks infinitival 'to'? As people, we know that this might reflect a typo, a mistake by a non-native-speaker, or a transcription error by an automatic speech recognizer; moreover, we might think it trivial to even think twice about this example. But for a machine, it is anything but trivial to determine whether *almost* matching lexically recorded expectations is good enough. For example, whereas *kick the bucket* can mean 'to die', *kick the buckets* (plural) cannot. So we do not want our agents to assume that all recorded constraints are relaxable – they have to be smarter about making related judgments.

Returning to the non-canonical "managed close the door", let us assume that the agent already processed the canonical input *Charlie managed to close the door* and stored the results in the TMR repository. Let's assume further that the new input is *The fire chief managed close the door,* for which the external parser we use, from the CoreNLP toolset (Manning et al. 2014), does not recognize that *close the door* is intended to be an xcomp. So our agent cannot directly align the parser output with the xcomp expected in the lexical sense for *manage*. As before, the agent can use the TMR repository as a search space and look for approximate string-level matches of "*managed close the door*". If it finds "managed to close the door," it can judge the similarity between the stored and new text strings and, if close enough, use the stored analysis to guide the new analysis. The natural extension is to relax the notion of similarity beyond surface string matching. The first level of relaxation might be to replace 'close' by any verb and 'the door' by an NP referring to any PHYSICAL-OBJECT, generating the following search query: **manage + $\mathbf{V_{BARE}}$ + $\mathbf{NP_{PHYSICAL\text{-}OBJECT}}$.** This would match stored inputs like *She managed to drink the espresso in 5 seconds flat*, whose associated stored TMR would provide the needed clue for how to combine the meanings of *manage* and *close* in our syntactically corrupted input. However, if this first level of relaxation fails to match a stored input, an even more relaxed pattern would remove the semantic constraint from the direct object, resulting in **manage + $\mathbf{V_{BARE}}$ + NP,** which would match inputs like *The tallest guy managed to win the race* ('race' is semantically an event, not an object), and would serve equally as a point of analogy for our *manage close the door* example.

Challenge 3. Literal vs. metaphorical meanings. Many expressions can be used in a literal or a metaphorical meaning, with the overall context being the only clue for disambiguation. For example, outside of contexts involving war, gangs or mafias, *I'm going to kill him!* typically indicates anger or, at most, the intention to punish someone for having done something undesirable. Similarly common are the metaphorical uses of *I hit him hard* and *I'm totally drowning!* We believe that the best way to prepare intelligent agents to analyze conventional metaphors (and most metaphorical usages are, indeed, conventional) is to record them in the lexicon. But runtime disambiguation, then, remains an issue. A good heuristic will be to simply check the TMR repository and see whether there is a frequency difference between the metaphorical and non-metaphorical usages, which should be the case, e.g., for *I'm going to kill him!*

Challenge 4. Exaggerations. People exaggerate all the time. (Get it?!) *Grandma drinks 20 cups of tea a day. If you go ½ mile-an-hour over the speed limit on that street they'll give you a ticket. Being a musician is tough, you earn like $1,000 a year.* Intelligent agents need to recognize exaggerations, which can be hyperboles or litotes, and convert them into their respective abstract representations which, in English, can be conveyed as *drinking a whole lot of tea, going slightly over the speed limit*, and *earning very little money.* The most direct way for an agent to detect an exaggeration is to compare the stated value with expectations stored in the ontology. For example, if the ontology says that people are generally not more than 7 feet tall, then saying that someone is 20 feet tall is surely an exaggeration. However, an ontology can be expected to cover typical heights of people, it very well might not cover things like "normal daily beverage consumption," "minimal speed infraction for getting a ticket" or "normal income range per year" – especially since the latter can differ greatly across different cultures and social groups. For these cases, the TMR repository can be of help.

The TMR repository should contain *interpretations,* not literal renderings, of inputs, for which many kinds of reasoning can be needed. For example, the agent must reason about non-literal language ("You're a pig" does not introduce an instance of the animal PIG into the context), about indirect speech acts ("Can you pass the pepper?" is a request not a question), as well as about exaggerations ("Grandma drinks 20 cups of tea a day" means she drinks a whole lot of tea). Focusing on exaggerations, the correct, interpreted TMR that is stored in an agent's TMR repository should reflect the conversion of an exaggerated scalar value into the highest – or, for litotes, lowest – value on the abstract scale. Compare the basic and reasoning-enhanced TMRs for our tea example, shown in Table 1.

Table 1. The basic and reasoning-enhanced TMRs for "Grandma drinks 20 cups of tea a day". The reasoning-enhanced TMR converts the exaggerated value into an abstract one.

Basic TMR		Reasoning-enhanced TMR	
DRINK-1		DRINK-1	
AGENT	GRANDMOTHER-1	AGENT	GRANDMOTHER-1
THEME	TEA-BEVERAGE-1	THEME	TEA-BEVERAGE-1
CONSUMPTION-RATE	**20 (MEASURED-IN CUP-PER-DAY)**	**CONSUMPTION-RATE**	**1**
TIME	find-anchor-time	TIME	find-anchor-time
TEA-BEVERAGE		TEA-BEVERAGE-1	
QUANTITY	**20 (MEASURED-IN CUP)**	**QUANTITY**	**1**

The reasoning-enhanced TMR asserts that drinking 20 cups a day is an exaggeration by aligning a tex-string value of "20 cups" with the "QUANTITY 1," the maximum value on an abstract scale from zero to 1. (If the ontology does not contain any relevant clues to guide this reasoning, it will need to be provided manually by the knowledge acquirer who is validating the results of the automatically-generated TMRs; see the discussion in the next section.) Once the agent detects this mismatch, it can use it to automatically enhance its ontology with a corresponding generalization: normal tea consumption is considerably less than 20 cups per day. Of course, we don't know how much less since the speaker of the exaggeration could have said anything from 20 to 1000 to a million cups a day. But, even though not maximally specific, this information can still be useful for reasoning about future inputs that include tea consumption. For example, if the agent subsequently receives the input "Joe drank 50 cups of tea yesterday", it can automatically – i.e., with no human intervention this time – hypothesize, with high confidence, that this is an exaggeration and automatically carry out the conversion from a specific scalar value to an abstract value.

Challenge 5. Elliptical and fragmentary utterances. Natural language text is full of elliptical and fragmentary utterances, some of which are stereotypical: *More cream, please* is a request for more cream; *Scalpel!* is a demand to be handed a scalpel; and *Anything else?* asks whether the speaker can give, or help the interlocutor with, anything else. One of the ways an agent can analyze these is by referring to ontological scripts – a method that is similar to the ontology-checking method of determining the normal range of heights for humans discussed above. So, if a robotic agent is configured to hand a surgeon implements during surgery, it will have the expectation that the physician's mention of a tool is a request to be handed the implement. (We are, of course, simplifying the actual complexity of the knowledge structures and the reasoning involved for reasons of space.) However, if the agent does not have recorded ontological scripts for a given domain it can still use the TMR repository to hypothesize what elliptical utterances might mean. For example, if the reasoning-enhanced TMR for "Scalpel!" is

```
REQUEST-ACTION-1
    AGENT          HUMAN-1      ; the speaker
    BENEFICIARY    HUMAN-2      ; the interlocutor
    THEME          TRANSFER-POSSESSION-1
TRANSFER-POSSESSION-1
    AGENT          HUMAN-2
```

BENEFICIARY HUMAN-1
THEME SCALPEL-1

then the agent can use this as a template for analyzing such inputs as “Coffee!” and “Another pickle, please!” Moreover, if the agent leverages this reasoning rule successfully – as determined by the fact that the TMR for “Coffee!” that it automatically generates is judged by a person to be correct – it can posit a generalization such as: When a sentence contains a mention of a PHYSICAL-OBJECT in isolation, or with the addition of select adverbs (e.g., please, now, here), this is a request to be given the object, as formally represented in a TMR structure like the one shown above. This generalization cannot be stored in the ontology like our generalization about quantities of tea because it relates not just to concepts but also to linguistic realizations of utterances, which are not within the purview of ontology.

5 Creating and Using a High-Quality TMR Repository

Because of all of the challenges listed above – as well as many more that we did not mention – it is difficult within the current state of the art for agents to generate perfect, reasoning-enhanced TMRs fully automatically. The most obvious way of creating the kind of high-quality TMR repository we have been talking about is for people to manually vet and, if necessary, correct TMRs automatically generated by the agent in the course of its normal operation. To return to an earlier example, if the agent has no information at all about normal tea-drinking practices, it has no way to know that 20 cups a day is an exaggeration: a person would have to correct the basic TMR that refers to the 20 cups, editing it to refer to the abstract value (QUANTITY 1). Only then can the agent detect future exaggerations in this realm.

Our past experience has shown that, given the appropriate tools, the process of semi-automatically creating gold-standard TMRs is not prohibitively difficult or time-consuming, and it is much faster than creating TMRs from scratch by hand. Although a gold-standard TMR repository has clear uses for intelligent agents configured within the OntoAgent architecture, it could also be viewed as a semantically deep alternative or supplement to traditional corpus annotation efforts, as we have discussed previously (McShane et al. 2005b).

6 Conclusion

We conclude by directly commenting on the issues posited to guide the crafting of submissions for this workshop.

Novelty: To the best of our knowledge, the proposed approach to using stored semantic interpretations of real texts to support the runtime analysis of new inputs is novel. However, it builds upon several existing theories and technologies: the theory of Ontological Semantics and its implementation in the OntoSem2 language processing system; the lexical and ontological knowledge bases used in that system; the OntoAgent cognitive architecture; and past work on compositionality, multiword expressions, and reasoning by analogy, among others.

Status of the Implementation: The theory of Ontological Semantics has recently been given a new implementation, called OntoSem2. It differs from its predecessor, OntoSem (McShane et al. 2016), in that it analyzes inputs incrementally and uses an architecture that is not a strict pipeline: e.g., reference resolution can be carried out before semantic analysis when applicable. OntoSem2 can already generate TMRs in fully automatic mode, though not yet for as broad a range of contexts as its predecessor. We are currently implementing the TMR repository that we have been discussing. The reasoning processes for dynamically leveraging the TMR repository have not yet been implemented. Although they will require more detailed specifications than have been provided here, formulating those specifications is a workaday task, as we understand the problem space well.

Benefits. Deep-semantic analysis will permit intelligent agents to effectively and naturally communicate with people during task-oriented collaboration. Agents must understand their task, their world, and how their interlocutors’ utterances fit into their mental model in order to function with human-like proficiency.

Limitations. The limitations of this approach to language analysis are that, at least at present, it is best suited to agent applications that focus on specific domains, rather than applications that cover all possible domains. For example, it makes sense for an agent to learn ontological facts about tea consumption only if it operates in a domain where that is important – e.g., as a medical assistant or an addiction counselor; and we can expect that human time in correcting TMRs will be best spent only if correcting TMRs in a specific domain of interest. The results of such human intervention can then inform language processing and reasoning in that same domain.

Benefit/Limitation Analysis. If we want intelligent agents to use language with human-like proficiency, we need to provide them with the same types of knowledge and resources as humans seem to use. Most of the NLP community has judged that achieving human-level proficiency is not sufficiently important to merit the requisite development time. We disagree, believing that people will not be satisfied with what knowledge-lean NLP can offer for very much longer.

Competing Approaches. The competing approaches are knowledge-lean NLP, along with its useful near-term applications that do not, however, support human-level reasoning by intelligent agents.

Next Steps. The next steps in making OS text processing useful are: a) continuing to develop the analysis engine; b) operationalizing notions like "sufficiently-close matching" (of new inputs to stored analyses) and reasoning by analogy; c) creating a large TMR repository; and d) testing all of these capabilities in demonstration systems with virtual agents and robots – all of which is in progress.

Outside Interest and Competing Approaches. Few groups are working on automating deep-semantic natural language understanding. Much of computational semantics is currently devoted to corpus annotation and the supervised machine learning that it supports; but the depth of those annotations is, in most cases, significantly less than what we describe here.

New Information About Language Functioning. We hypothesize that people have, and use, the equivalent of a TMR repository during language understanding. For example, if you hear just "The cat ran out…" what do you predict comes next? Perhaps *the door, of the room, of the yard, into the street*? It is likely that *something* comes to mind, and that something derives from ontological knowledge about the world as well as past experience with individuals in it. This paper has described how we have tried to lasso that knowledge into a form that is usable by intelligent agents.

Finally, given our collective decades-long experience working on these issues, we do not underestimate the number of devils in the details of operationalizing analogy detection, approximate matching, and so on. However, we have treated countless such devils before and have come to believe that they are, almost always, quite benign – they just require close attention and dedicated conceptual and descriptive work.

Acknowledgments

This research was supported in part by Grant N00014-16-1-2118 from the U.S. Office of Naval Research. Any opinions or findings expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research. Our thanks to Igor Boguslavsky for his insightful commentary on a draft of this paper.

References

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations,* pp. 55–60. Stroudsburg, PA: The Association for Computational Linguistics.

McShane, M., Nirenburg, S. and Beale, S. 2005a. An NLP lexicon as a largely language independent resource. *Machine Translation*, 19(2): 139-173.

McShane, M., Nirenburg, S., Beale, S. and O'Hara, T. 2005b. Semantically rich human-aided machine annotation. *Proceedings the Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*, pp. 68-75, at the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05). Stroudsburg, PA: Association for Computational Linguistics.

McShane, M. and Nirenburg, S. 2012. A knowledge representation language for natural language processing, simulation and reasoning. *International Journal of Semantic Computing*, 6(1): 3-23.

McShane, M. and Nirenburg, S. 2015. Decision-making during language understanding by intelligent agents. *Artificial General Intelligence*, Volume 9205 of the series Lecture Notes in Computer Science, pp. 310-319.

McShane, M., Nirenburg, S. and Beale, S. 2015. The Ontological Semantic treatment of multiword expressions. *Lingvisticæ Investigationes*, 38(1): 73-110. John Benjamins Publishing Company.
McShane, M., Nirenburg, S. and Beale, S. 2016. Language understanding with Ontological Semantics. *Advances in Cognitive Systems* 4:35-55.
Nirenburg, S. and Raskin, V. 2004. *Ontological Semantics*. Cambridge, Mass.: The MIT Press.