

МНОГОСТОРОННИЙ ПОДХОД К РЕФЕРЕНЦИИ В АНГЛИЙСКОМ И РУССКОМ ЯЗЫКАХ

МакШейн М. (mcsham2@rpi.edu)

Политехнический институт Ренсселера,
Трой, Нью-Йорк, США

Ключевые слова: семантический анализ, обработка референции, эллипсис

A MULTI-FACETED APPROACH TO REFERENCE RESOLUTION IN ENGLISH AND RUSSIAN

McShane M. (mcsham2@rpi.edu)

Rensselaer Polytechnic Institute, Troy, NY, USA

This paper argues that the detection and resolution of referring expressions can be profitably distributed across modules of a language processing system, rather than being bunched at the end of a text analysis pipeline. The approach is being implemented within the OntoAgent cognitive architecture, which supports the development of multi-functional, language-endowed agents that can collaborate with people in task-oriented applications. Although current development within OntoAgent orients around English, the architecture itself and most of its knowledge bases are language-independent. Drawing upon my past descriptive work on reference and ellipsis in Russian, I will suggest how the same reference resolution strategies might be applied to this and other languages. More generally, I will motivate the need to approach linguistic phenomena in a holistic paradigm, rather than as highly compartmentalized subtasks, which has become the norm for natural language processing applications.

Keywords: semantic analysis, reference resolution, ellipsis

1. Introduction

A common approach in scientific research and technological development is the “divide and conquer” analytical approach, in which the study of naturally-occurring phenomena is divided into subfields and subproblems treated by different research communities, often within different R&D paradigms. Despite well-known scientific and practical motivations for this methodology, there are also drawbacks. For example:

- (1) It is often assumed that the prerequisites needed to support the treatment of a (sub)problem are, or will somehow become, available, even if that expectation is unrealistic. In fact, fulfilling the necessary prerequisites is often more difficult than solving the problem itself.
- (2) It is assumed that when external prerequisites become available, the solution for the selected subproblem will still be valid. Such speculation is particularly questionable in the case of challenging prerequisites, since the knowledge and processing needed to fulfill them might offer a more natural solution to the original problem.
- (3) The solutions for subproblems will ultimately converge into a solution for the full problem. This requires integrating different theories, knowledge bases and input-output expectations of processors, a noble goal fraught with practical, scientific and sociological snares.
- (4) It is assumed that some narrowly delineated problem space is a reasonable proxy for the actual problem space, i. e., that methods that are shown to work on a non-real-world problem can be successfully, and without too much additional effort, applied to the solution of the associated real-world problem.

Within the realm of reference resolution, the “divide and conquer” approach has resulted in R&D paradigms that, in my opinion, fail with respect to all of the above points. For example, in Anglo-centered natural language processing (NLP), reference resolution has widely been treated as a machine learning problem specified using the following, oversimplifying rules of the game:

- (1) It is expected that manually annotated corpora will be provided for the training and evaluation stages of reference resolution engine development: i. e., reference engines receive as input a perfect syntactic parse of sentences in naturally occurring text, even though achieving this automatically is far beyond the current state of the art.
- (2) Referring expressions of interest, called “markables”, are manually selected, so the detection stage of reference processing is (artificially) removed.

- (3) Markables do not cover all the types of referring expressions—they are limited to cases that lend themselves to resolution using machine learning methods. Specifically:
- a. Markables must be noun phrases: referring verbs are excluded.
 - b. Markables must be overt: ellipsis is excluded.
 - c. Markables must have overt noun phrase antecedents: verbal, clausal, multi-clause, elided and extra-linguistic antecedents are excluded.
 - d. The antecedents for markables must be contiguous: dynamically composed sets that can serve as an antecedent for plural referring expressions are excluded.
 - e. The antecedents for markables must be unambiguous: if annotators are expected to have difficulty agreeing on the antecedent, the given referring expressions is excluded.

Clearly, this problem space—described in far more detail in the Message Understanding Conference (MUC) co-reference task specification (Chinchor 1997)—represents only a fraction of reference resolution challenges present in natural language texts (see McShane 2009 for details).¹ Although work in this paradigm has led to improvements in machine learning methods themselves, I would suggest that this invented problem space has little to do with the actual problems faced by text analysis systems: taking raw text as input and arriving at a full semantic analysis, which necessarily includes the detection and resolution of overt and elided referring expressions. When one addresses the latter problem within an end-to-end language processing system, not only do the challenges look different, so do the available solutions to overcome them.

In this paper, I describe how the detection and resolution of referring expressions is being distributed across modules of the OntoAgent text analysis system. In OntoAgent, individual phenomena are treated as soon as the necessary heuristic evidence becomes available. Our overarching development strategy is to strive for theoretical and practical progress over the long term on the fundamental issues of semantic analysis. As a result, a) we do not expect prerequisites to be fulfilled externally, by systems or human input outside OntoAgent; b) there is no inherent ceiling on the quality of OntoAgent results, and c) we can (and do) exploit OntoAgent in applications even before it reaches its maximum capabilities. Although current applications of OntoAgent orient around English, the architecture itself and most of the supporting knowledge bases are language-independent. Drawing upon my past descriptive work on reference and ellipsis in Russian, I will suggest how the same reference resolution strategies might be applied to this and other languages.

¹ This task, and the manually annotated corpus developed for it, were used to support government-sponsored competitions.

2. Modules of Text Processing

Typically, natural language processing (NLP) systems are implemented as pipelines, such as the one shown in Figure 1.²



Fig. 1. A typical NLP pipeline

Although pipeline architectures offer the well-known advantages of modularity, they also suffer from lost potential. For example, key aspects of syntactic analysis, such as decisions about prepositional phrase attachment, cannot be made without semantic input,³ and key aspects of semantic analysis, such as lexical disambiguation, require reference resolution.⁴ In fact, psycholinguistic evidence—which informs but does not constrain our cognitive modeling decisions—shows that people attempt to resolve reference (as shown in eye-tracking experiments) as soon as possible upon hearing elements of input (Tanenhaus 1995). So the “simplifications” promised by a modular architecture can inadvertently lead to the unnatural isolation of phenomena, blocking the necessary, bidirectional passing of heuristic information across modules.

Within the OntoAgent cognitive architecture (described in Section 3), we try to balance the practical utility of a pipeline architecture with the linguistic realities that defy strict compartmentalization. One aspect of this balancing act is distributing the treatment of reference as shown in Figure 2. RR1-RR3 are reference resolution engines that fire before the main reference resolution module. The output of each of these engines can be used to inform all downstream processing.

The cartoon eyeballs in Figure 2 emphasize that the NLP capabilities discussed here are embedded in a more comprehensive agent architecture. This architecture centrally includes static knowledge resources (lexicon, ontology), all manner of processing engines (for simulation, reasoning, NLP), dynamically populated agent memory, and components we will not address here, such as a physiological simulation representing the agent’s body. Let us zoom out to this big picture of agent modeling before returning to our topic at hand.

² Of course, many practical systems do not include all the modules illustrated in the diagram.

³ For example, in *Shirley hit the boy with the horn*, did Shirley hit a boy holding a horn, or did she use a horn as the instrument with which to hit the boy? This cannot be determined without contextual-semantic analysis.

⁴ See McShane et al. 2010 for a discussion of how reference resolution can be used to inform lexical disambiguation.

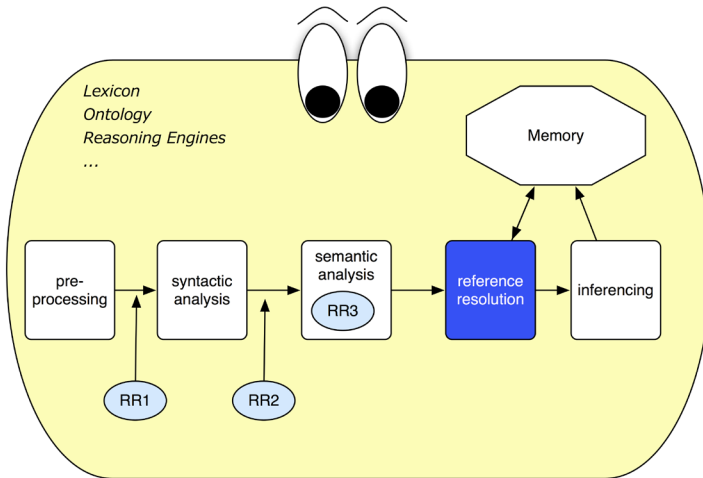


Fig. 2. OntoAgent distribution of reference treatment across processing modules. RR1-RR3 are reference processing engines that are applied before the main “reference resolution” module of text processing.

3. The OntoAgent Environment

OntoAgent is a knowledge-based intelligent agent environment inspired by the traditional goals and motivations of artificial intelligence: attempting to achieve human-level behavior by modeling agents with human-like capabilities of perception, reasoning and action. OntoAgents include integrated physiological and cognitive simulations, with the latter centrally including natural language processing capabilities. A recent prototype application area is Maryland Virtual Patient (Figure 3), a clinician training system in which a cohort of virtual patients can be diagnosed and treated in open-ended, interactive simulations that also optionally include an automatic tutor and additional virtual medical personnel (McShane et al. 2007, 2012 a, b, 2013 a, b; Nirenburg et al. 2008).

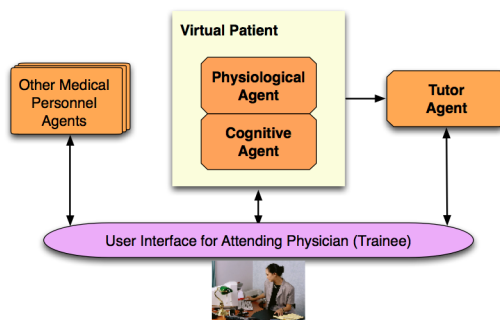


Fig. 3. The Maryland Virtual Patient application

The goal of language understanding in OntoAgent is for the agent to achieve human-level understanding of input text and use that knowledge to populate its memory. Stored memories then support subsequent reasoning and action. In this infrastructure, reference resolution is defined in terms of agent memory: any object or event referred to in a text could be *new* to the agent, in which case a new anchor in memory must be created, or it could be already *known* to the agent, in which case the new information should be appended to the existing anchor. Memory augmentation of this type is presumably what humans achieve, and therefore what intelligent agents must emulate, when processing language input, regardless of whether or not the approaches used in software development show any similarities to the operations of human wetware.

4. Semantic Analysis as the Substrate for Reference

Since reference resolution applies to memory, and since agent memories are stored as ontologically-grounded, disambiguated representations of text meaning, we must begin by briefly describing what we mean by semantic analysis (Nirenburg and Raskin 2004; McShane 2009). Consider example (1), in which [e] indicates an elided category.

- (1) Yesterday Sasha played like crazy and [e] fell asleep by eight o'clock.
Вчера Саша играла как зверь и [e] заснула к восьми часам.

This example, like any other, can only have a real-world meaning within some context: i. e., there must be a particular Sasha who, on a particular day, carried out a particular event of playing hard and who was subject to a particular instance of falling asleep at the appropriate interpretation of 8 o'clock, be it a. m. or p. m.

To ground this example and our further discussion of it, let us assume that I am telling this to my mother over the phone. She will know: (i) that Sasha is one of my dogs; (ii) that when she plays she does things like run, fetch her tennis ball, chase squirrels, and wrestle with her canine brother; (iii) which day I'm talking about, by calculating *yesterday* based on the day of the phone call; (iv) that I'm talking about 8 p. m. because only 8 p. m. can occur after a day's worth of playing; and (v) that this is reportable news because Sasha usually doesn't fall asleep by 8:00. If we were to configure a "Mom" intelligent agent with the same background knowledge and reasoning ability as its human counterpart, I would want it to reach these same interpretations.

Two core aspects of arriving at a full semantic interpretation are lexical disambiguation and the establishment of semantic dependencies (Beale and McShane, in preparation). Both of these processes are supported by a text analysis system that relies on a highly detailed computational lexicon and a language-independent ontology. The lexicon includes linked syntactic and semantic expectations for argument-taking words. For example, two of the many senses of the verb *play* are shown below, using a simplified formalism.

```

play-v1
def          to play a musical instrument
example     John is playing (his cello).
syn-struct
  subject    $var1
  v          $var0
  directobject $var2 (opt +)
sem-struct
  PLAY-MUSICAL-INSTRUMENT
    AGENT    ^$var1 (default HUMAN)*
    THEME    ^$var2 (default MUSICAL-INSTRUMENT)*

play-v2
def          of dogs—to play (fetch balls, run, wrestle, etc.)
example     Spot is playing in the backyard.
syn-struct
  subject    $var1
  v          $var0
sem-struct
  PLAY-DOG
    AGENT    ^$var1 (default DOG)*

```

Play-v1 is optionally transitive and maps to the ontological concept `PLAY-MUSICAL-INSTRUMENT`. The meaning of the subject (in the basic diathesis) maps to the `AGENT` case-role, whereas the meaning of the direct object (in a basic diathesis) maps to the `THEME` case-role.⁵ The ontology specifies that the `AGENT` of `PLAY-MUSICAL-INSTRUMENT` should be `HUMAN`, and that the `THEME` should be `MUSICAL-INSTRUMENT`. The semantic constraints are marked by an asterisk because they are not actually listed in the lexicon since they are accessible in the ontology. The system will select this sense only if all syntactic and semantic constraints are met. *Play-v2*, by contrast, is an intransitive sense that maps to `PLAY-DOG`. It can be selected by the text analyzer only if the subject—to be realized as the `AGENT`—is of the semantic type `DOG`.

A core aspect of *OntoAgent* text processing is that it is largely language independent. In fact, its approach to knowledge-based language processing, which implements the theory of Ontological Semantics (Nirenburg and Raskin 2004), was originally developed for interlingual machine translation. So large portions of a lexicon formulated this way can be ported across languages with only minimal editing required (McShane et al. 2005). For example, Russian equivalents for our examples can be created by simply changing the headwords to `играть-v1` and `играть-v2`. These senses, if processed by a Russian system analogous to our current English one, will generate the same text meaning representations for semantically equivalent input. All subsequent reasoning and action by *OntoAgents* will then be identical.

⁵ Non-basic diatheses are treated using syntactic transformations in the analyzer.

The second core static knowledge base is the ontology, which is completely language-independent. Consider a subset of the property values used to describe the concept **PLAY-DOG**, which describes the special ways that dogs—as opposed to people or horses—play.

```

PLAY-DOG
AGENT      DOG
EFFECT     HAPPINESS (RANGE (> .8))
HAS-EVENT-AS-PART  PLAY-FETCH  (THEME      BALL, STICK, FRISBEE)
                                     CHASE      (THEME      SQUIRREL)
                                     DIG        (INSTRUMENT PAW)
WRESTLE-DOG
RUN
SNIFF      (THEME NATURAL-OBJECT)
    
```

This frame says that dogs are the agents of **PLAY-DOG**; that the effect of playing is making the dog happy; that typical subevents include fetching a ball, stick or Frisbee, chasing squirrels, digging, wrestling with other dogs, running around, and sniffing the great outdoors. Due to space constraints, many details of ontology form and content omitted; the main point is that the ontology contains key aspects of a person’s knowledge about the world, which an intelligent agent should also possess in order to understand language robustly.

The automatic analysis of text, which relies on lexicon and ontology, results in the generation of text meaning representations, which are comprised of numbered instances of ontological concepts connected by ontological relations. The actual numbering of concept instances depends upon the past processing history of a given agent. So, at a given point in its simulated life, an agent (whether processing English or Russian input) might generate the following text meaning representation for our example. Explanatory comments are introduced by semi-colons.

Yesterday Sasha played like crazy and fell asleep by eight o’clock.

PLAY-DOG-435

```

AGENT      DOG-27
INTENSITY  1 ; on the abstract scale {0,1}
ABSOLUTE-TIME  MONTH 3
                                     DAY 13
                                     YEAR 2014
RELATIVE-TIME  < FALL-ASLEEP-271 ; “<” indicates “before”
    
```

FALL-ASLEEP-271

```

EXPERIENCER  DOG-27
ABSOLUTE-TIME  HOUR 20
                                     MONTH 3
                                     DAY 13
                                     YEAR 2014
    
```


RELATIVE-TIME > PLAY-DOG-435
DOG-27
 HAS-PERSONAL-NAME Sasha

The concepts themselves and their relationships to other concepts are drawn directly from the sem-struct zones of lexical senses and are combined using the Hunter-Gatherer constraint-based semantic analysis engine (Beale 1996).

This text meaning representation reflects approximately what we expect a human to arrive at when contextually interpreting this input. However, the work of sentence processing is not yet done: the new information has to be incorporated into human or agent memory. Below is a very partial view of what agent memory might look like after hearing this sentence. My human mother—or our “Mom” agent—has an anchor for Sasha that could include hundreds or thousands of pieces of information. The newly reported events in boldface are dynamically added to existing memories. They are also recorded as the heads of their own frames in memory, supplied with all of the extra information available in the text meaning representation above (e. g., absolute and relative times of the events).

DOG-27
 HAS-PERSONAL-NAME SASHA
 HAS-FAMILY-NAME McSHANE
 HAS-OWNER HUMAN-88
 COLOR BLACK, TAN
 WEIGHT 63 (MEASURED-IN POUND)
 HAS-BIRTHDATE (MONTH 8) (DAY 13) (YEAR 2009)
 AGENT-OF INGEST-71, CUDDLE-889, FETCH-204, DIG-336
 [many more events], PLAY-DOG-435
 EXPERIENCER-OF STROKE-ANIMAL-44, GROWL-33, *[many more events]*,
 FALL-ASLEEP-271

Population of agent memory is the culmination of contextually grounded semantic analysis, including reference resolution. Earlier, we described, albeit very briefly, how our intelligent agents generate basic text meaning representations, which includes lexical disambiguation and the establishment of semantic dependencies. The next section describes, in slightly more detail, how the processing of reference is distributed across processing modules. Due to constraints of time and space, discussion will focus on those aspects of reference processing that are treated *before* the main reference resolution engine is called.

5. Reference Resolution across Modules

In this section, I describe each of the engines RR1-RR3 (cf. Figure 2) in turn and how they contribute to overall text analysis in our English system. I will also suggest how reference phenomena found in Russian—and, by extension, any other

language—might be treated using the same methods. I should emphasize that I am not orienting around the output of any particular Russian text processors. Instead, I am making certain assumptions about (a) the kinds of inputs and outputs typical of preprocessors and syntactic analyzers cross-linguistically, and (b) the kinds of phenomena that tend to be most challenging for those engines.

5.1. Reference Resolution Engine 1

OntoAgent text processing begins with preprocessing, for which we use the Stanford preprocessor (Klein and Manning 2003a,b). Among its many functionalities are HTML mark-up stripping, sentence and word boundary detection, part of speech tagging, morphological analysis, and certain aspects of named entity recognition. The latter contributes to reference resolution since named entities are referring expressions. However, although the Stanford preprocessor groups named entities, it does not semantically analyze them. It does, however, provide useful input to an OntoAgent engine—a component of RR1—that does semantically analyze them. For example, for the input string *Army Capt. Patrick Horan*, the Stanford preprocessor returns the structure on the left, which OntoAgent uses—along with an onomasticon (lexicon of proper names) and associated rule set—as input when generating the structure on the right:

(NP	HUMAN-1	
(NNP ARMY)	HAS-TITLE	ARMY CAPT.
(NNP CAPT.)	HAS-PERSONAL-NAME	PATRICK
(NNP PATRICK)	HAS-SURNAME	HORAN
(NNP HORAN))		

Why resolve this meaning so early, before “mainline” semantic analysis kicks in? First, because the necessary heuristic evidence is already available; second, because this analysis can then serve as an “island of confidence” for later clause-level semantic analysis. That is, knowing that this individual is HUMAN will help to disambiguate the verb in the clause for which it fills a case role. Named entities in Russian could be treated in the same manner, given a Russian onomasticon and rules about named entity formation comparable to those available for English.

The second aspect of reference processing that requires only the results of preprocessing—along with static knowledge bases that are always accessible—is the detection of certain kinds of ellipsis. For example, when a modal or auxiliary verb occurs before a hard discourse break signaled by a period, colon, semi-colon or question mark⁶, this almost always indicates an elided verb, as illustrated by (2):

- (2) Zhenya managed to get here on time but Alla couldn't [e].
Женя смог добраться сюда вовремя, а Алла не смогла [e].

⁶ Commas are not high-confidence predictors due to widespread cases like the following: Brian didn't, or at least didn't seem to, want to go to Aruba.

Although the inventory of relevant modals/auxiliaries differs across languages, the ellipsis-detection generalization remains the same (McShane et al. 2012d). Moreover, in a highly elliptical language like Russian, it might be useful to seek additional high-frequency, high-confidence elliptical patterns and posit the associated underlying verb. For example, sentences like (3) and (4) could be recognized as elliptical using patterns that refer only to parts of speech (remember, we have not launched syntactic analysis yet).

- (3) Noun + preposition + noun + ./? → Noun + [verb] + preposition + noun + ./?
 Я в магазин. → Я [e=verb] в магазин.
 I to store → I [e=verb] to store
 ‘I’m going to the store.’
- (4) Noun + adverb + ./? → Noun + [verb] + adverb + ./?
 Ты сразу? → Ты [e=verb] сразу?
 You immediately? → You [e=verb] immediately?
 ‘Are you going to do it <come, go, etc.> right now?’

Detecting ellipsis at this stage and positing an underlying verb—even though the system can’t know what it means yet—should, presumably, improve syntactic analysis.

5.2. Reference Resolution Engine 2

The next stage of OntoAgent processing is syntactic analysis, for which we use the Stanford syntactic dependency analyzer (de Marneffe et al. 2006). From the parse, OntoAgent detects several types of structures that are potentially elliptical and adds reference-oriented metadata to the current state of analysis to support downstream processing. Two elliptical phenomena that can be detected at this stage are gapping and unexpressed subjects of VP coordinate structures, as illustrated by examples (5) and (6):

- (5) Lori had a sandwich and Mary [e], a salad.
 Лори съела бутерброд а Мэри [e] салат.
- (6) Tom had a sandwich and [e] went to work.
 Том съел бутерброд и [e] пошел на работу.

We will concentrate on the example of gapping, though subject ellipsis in these constructions is treated analogously.

The Stanford parser typically analyzes gapping structures as conjoined nominals—essentially, appositives. Simplifying a bit, the output structure for the 2nd half of (5) is: (NP (NP Mary) (NP a salad)). Although this output is incorrect—i.e., the ellipsis is not detected—it is *consistently* and *predictably* incorrect, and this provides us with a useful heuristic. The RR2 engine includes an inventory of clause-level heuristics to evaluate whether a given output with this structure is actually an appositive

or an undetected gapping configuration. If it concludes it is the latter, then the engine: (1) recovers the missing verbal element by copying the verbal string (not yet semantically analyzed!) from the previous conjunct and (2) adds metadata to the copied string that explicitly blocks instance-coreference—i.e., the events in question are of the same type but with different case-role fillers. This revised syntactic output greatly reduces ambiguity in later processing of semantics and reference.

Since Russian widely employs both of these elliptical strategies, this approach to recovering the elided material should be equally applicable—of course, assuming that Russian parsers, like English ones, do not have alternative, high-confidence methods of detecting elided categories.

5.3. Reference Resolution Engine 3

The next stage of processing is basic semantic analysis, which results in text meaning representations like the one described above. The two aspects of reference processing carried out at this stage are (1) the lexically-supported detection of non-referring expressions and (2) the detection and resolution of certain kinds of ellipsis. We will consider each in turn.

Lexically-supported detection of non-referring expressions. Many nouns and verbs are not referring expressions and, therefore, should not be subject to reference resolution procedures. Examples of non-referring expressions in English include, non-exhaustively, pleonastic *it* (*It's cold in here!*); auxiliary uses of polysemous verbs (*I have already finished*); and non-compositional elements of idioms (*He kicked the bucket, meaning 'He died'*). The OntoAgent lexicon includes lexical senses for words and expressions that define the contexts in which they are non-referring and prepare the system to correctly analyze them.

Consider the lexical senses for the English idiom *kick the bucket* and the roughly equivalent Russian *сыграть в ящик*.

```

kick-v4
  def          to die (colloquial, humorous)
  example     Old Mr. Jones kicked the bucket.
  syn-struct
    subject    $var1
    v          $var0
    directobject $var2 (root: bucket (num: sing))
                (contains: the (PoS: article))

  sem-struct
    die
      experiencer ^$var1
    ^$var2      null-sem +

сыграть-v4
  def          умереть (colloquial, humorous)

```

```

example    Старик Иванов сыграл в ящик.
syn-struc
  subject  $var1
  v        $var0
  PP
    prep   $var3      (root: в)
    obj    $var2      (root: ящик (num:sing))
sem-struc
die
experienter ^$var1
^$var2 null-sem +
^$var3      null-sem +

```

The syn-strucs of both senses indicate which lexical elements must participate in the idiom, along with grammatical constraints on them: in English ‘bucket’ must be singular and must be used with the article ‘the’; in Russian, ящик must be singular and the preposition heading the prepositional phrase must be в. In both languages, the ontological meaning is DIE, and the EXPERIENCER of dying is realized as the subject of the clause. (Stylistic nuances are not central to this discussion and will not be pursued here.) In both languages, the actual words used to express the meaning ‘die’ are not referring expressions: there is no bucket, box, kicking or playing involved. The sem-strucs indicate that *bucket* and *box*, along with the preposition в in Russian, should not be productively analyzed using the descriptor “null-sem +”, which is an abbreviation for “null semantics”. As such, there will be no explicit trace of these words in the text meaning representations, and reference resolution will not be applied to them, which is exactly what is needed. In short, preparing the system to carry out basic semantic analysis of idioms also blocks the unwarranted search for coreferents for non-referring expressions.

Detection (and resolution) of certain kinds of ellipsis. Just as idioms can be automatically detected and resolved thanks to highly specified lexical entries, so can certain kinds of ellipsis. For example, when modal and aspectual verbs take a direct object that is ontologically an ОБЪЕКТ, this always indicates semantic ellipsis of the main verb, as in shown in (7) and (8).

- (7) Dima desperately wants [e] a dog / a hamburger / a bike.
 Дима страшно хочет [e] собаку / гамбургер / велосипед.
- (8) Anya finished [e] the article / the blanket only yesterday.
 Только вчера Аня кончила [e] статью / плед.

Dedicated lexical senses of modals and aspectuals *anticipate* verbal ellipsis in such contexts and launch calls to procedural semantic routines that attempt resolve the meaning of the unexpressed event. Consider, for example, the lexical sense for *finish* that would be used to analyze (8).

```

finish-v2
  def      to complete some action involving the direct object

```

example Stacy finished the book yesterday
 (elided “reading, writing, binding”, etc.)

syn-struct

subject	\$var1
v	\$var0
directobject	\$var2

sem-struct

refsem1	(sem EVENT)
AGENT	^\$var1
theme	^\$var2 (sem OBJECT)

meaning-procedure

seek-specification refsem1 (^\$var1 ^\$var2)

This sense is used only if \$var2 refers to an ontological OBJECT (if it referred to an event there would be no ellipsis, as in *John started washing the car*). When this is the case: (a) there must be an elided EVENT; (b) the meaning of the subject and direct object almost certainly fill the AGENT and THEME case roles, respectively, of that event; and (c) the actual meaning of that event must be dynamically computed based on the meanings of the subject and direct object. The engine that attempts to contextually compute that more specific meaning is triggered by a call to the function “seek-specification”, which is listed in the meaning-procedures zone of the entry. If this function can make a confident hypothesis regarding the actual meaning of the event, then that hypothesis is recorded in the text meaning representation—e.g., in the case of Anya finishing a blanket, the hypothesis most strongly suggested by ontological search should be KNIT or CROCHET. If the engine cannot confidently constrain the meaning of the event, then the underspecified EVENT remains in the meaning representation. Lexically-recorded procedural semantic routines are used widely in OntoAgent text processing, as described in McShane et al. 2004. The point here is that linguistic expectations about elliptical configurations can be lexically recorded and leveraged to support reference processing *prior* to the working of the dedicated reference module.

Russian employs ellipsis more widely than English, and ellipsis in many configurations can be resolved using highly predictable patterns (McShane 1999, 2000 a, b, 2005). This suggests that anticipatory lexicalization of the patterns could be profitably employed. For example, conjunction structures can involve the ellipsis of the 2nd conjunct’s subject and direct object; configurations anticipating such ellipsis can be lexically recorded using the conjunction as a headword.

и-conj12

def: pattern “subject verb direct-object и [e] verb [e]”

ex: Лиза купила пломбир и сразу съела.

Liza bought an ice cream and ate it right up.

comments: a coordinate configuration with ellipsis of the latter subject and direct object

```

syn-struct
  subject      $var1
  v            $var2
  directobject $var3
  и           $var0
  v            $var4
sem-struct
  CONJUNCTIVE-DISCOURSE-RELATION
  DOMAIN      refsem1
  RANGE       refsem2
meaning-procedure
analyze-clause refsem1 ($var2 (subject:$var1 DO:$var3))
analyze-clause refsem2 ($var4 (subject:$var1 DO:$var3))

```

This lexical sense converts syntactic configurations consisting of a subject, verb, direct object, the conjunction *и*, and another verb, in that order. Within OntoSem, modifiers are always permitted unless explicitly blocked, meaning that a sentence element like *сразу* ‘immediately’ in our example will be permitted and compositionally incorporated into the text meaning representation.

The sem-struct says that this configuration contains two clauses that are connected by a conjunctive discourse relation. Those clauses are referred to as *refsem1* and *refsem2*—essentially, pointers to reified structures. Without knowing in advance the particular words that will be used in an input, the lexicon entry cannot predict which concepts will be instantiated or their dependency structure. Those determinations must be made using the normal analysis procedures that are carried out by a pair of calls to the procedural semantic function “analyze-clause”.

The key to recovering the elided arguments in this configuration lies in the explicit indication of the arguments of the 2nd verb. Specifically, when the verb indicated by *\$var2* is being analyzed, it uses *\$var1* as its subject and *\$var3* and its direct object; and when the verb indicated by *\$var4* is analyzed, it *also* uses *\$var1* as its subject and *\$var3* as its direct object. In short, this lexical sense anticipates an elliptical structure and explicitly tells the analyzer how to resolve the reference of those elided elements.

Based on my past work on ellipsis in Russian and Polish (cited above), I believe that many elliptical patterns could be effectively treated using this pattern-based strategy. To bypass the rather dense formalism, I will present the patterns via language examples, leaving readers to construe their formal lexical specification independently. Among the relevant patterns are elliptical *нет* configurations (9), clausal conjunction with an elided 2nd direct object (10), multi-sentence ellipsis of subjects and objects (11), repetition structures, which are often used for emphasis or stylistic effect (12), the ellipsis of verbs of motion and speech (in which the missing verbs may or may not have been detected by RR2) (13), and many more.

- (9) Лори любит кататься на велосипеде, а Лиза нет.
Lori likes to ride her bike but Liza doesn't.

- (10) «В любом случае завтра, нет, уже сегодня, сменю замок. Здесь этим занимается сторож, куплю **что-нибудь сильно замысловатое**, а он [e] поставит» (Хмелевская).
In any case tomorrow, no, today I'll change the lock. The superintendant deals with those sorts of things. I'll buy something really elaborate and he'll install it.
- (11) «Я имел подлость убить сегодня **эту чайку**. Кладу [e] у ваших ног» (Чехов).
«Today I was so base as to kill this seagull. I lay it at your feet» (Chekhov).
- (12) «Красное небо, уже начинает восходить луна, и я гнала **лошадь**, гнала [e]» (Чехов).
Red sky, moon on the rise, and I drove that horse on, I drove it hard (Chekhov).
- (13) Я не [e] об этом, я [e] о другом.
I'm not talking about that, I'm talking about something else.

To recap, at this point in text processing, the system has generated text meaning representations which already include some reference-oriented decisions: certain elided categories have been detected and a subset of those have been resolved (others await future resolution procedures); in addition, some non-referring expressions have been detected and removed from further consideration by the main reference resolution engine.

Although space does not permit a full description of the interdependencies among processing modules in OntoAgent, one important detail must be mentioned. The lexical senses that support the types of analysis described above are actually leveraged *before* syntactic parsing as well. Specifically, the syntactic patterns recorded in the syn-struc zones of lexicon entries can be used to *force* certain decisions by the syntactic parser. This is particularly important in cases in which the lexically recorded patterns detect ellipsis because, if the parser fails to detect ellipsis, it can produce wildly erroneous output that defies effective semantic analysis. The reason for mentioning this detail out of order with respect to the basic pipeline is pedagogical: the only part of lexical senses important to the parser are syn-struc zones, but it would be strange and unmotivated to describe the syn-struc zones of phrasal lexical senses decoupled from the sem-struc zones of those same entries.

5.4. The Dedicated Reference Resolution Module

Although this paper is centrally about reference resolution, I will not spend much time describing the main reference resolution engine. There are several reasons for this decision: first, constraints of time and space must be observed; second, the associated theory and engine is described in detail in McShane 2012c, 2013c, and Submitted; and third, the real point of this paper is to suggest that reference processing, like any linguistic phenomenon, is best treated holistically rather than in a compartmentalized fashion. However, to avoid a gaping hole in this portrayal of reference processing, I will briefly encapsulate the workings of the dedicated reference module.

This module is called after the basic text meaning representations have been generated. Reference procedures apply to all instances of OBJECTS and EVENTS comprising those meaning representations (recall that all non-referring objects and events will have been excluded by now). The reference engine first determines whether a textual coreferent for an expression should be sought. Although this is straightforward for pronouns and indefinite referring expressions (pronouns always trigger the search for a textual coreferent whereas indefinite referring expressions never do), it is much more complex for verbs, definite referring expressions and named entities. An inventory of knowledge-based algorithms specific to each class of referring expression guides the agent's decision-making with regard to seeking and establishing textual coreference relations. Whether or not an entity has a textual coreferent, it must ultimately be anchored to the agent's memory in the way described earlier, drawing along with it all new information presented in the text. Deciding whether a mentioned entity links to an available anchor in memory or requires the establishment of a new one is carried out based on matching feature value of the input with feature values of available anchors. The success in automating this matching process is highly dependent upon the domain and application. For example, in the Maryland Virtual Patient application, the virtual patient can make certain assumptions about the scope of relevant entities in the world that greatly simplifies the process of memory population management. By contrast, if an OntoAgent is tasked to process a large amount of running text, the challenges of cross-textual reference resolution will surely skyrocket.

The reference resolution module is largely language-independent since its primary source of heuristic evidence is text meaning representations, which are written in the ontological metalanguage. Of course, some aspects of surface realization of language also provide reference clues, such as the distance between a referring expression and each of its candidate antecedents, and, for languages like English, the use of indefinite vs. definite articles. However, much of the heavy-duty reasoning related to memory population and management will be the same for agents operating in any language environment.

6. Concluding Thoughts

The narrow goal of this paper has been to suggest some advantages to distributing reference processing across modules of language processing. Since we are finding this approach useful for English, and since Russian not only shares some difficult reference phenomena with English but adds plenty more to the mix, it seems plausible that this approach might be useful for Russian as well.

However, setting aside the narrow problem of reference resolution, the overarching conclusion is that the big picture of language analysis should more centrally inform both the selection of subproblems by the NLP community and the approaches used to solve them. The "isolationist" mindset that drives much of the recent system building does not show much promise for solving the hard problems of NLP, despite its success in producing engines suitable for simpler tasks supporting limited applications in the near term.

I am not suggesting that the maximally deep, knowledge-heavy semantic analysis pursued by *OntoAgents* will produce high-quality results over open text in the near term—it certainly will not; after all, its success is predicated on finding solutions to some of the hardest problems in a variety of subareas of cognitive science. Nor am I suggesting that it is inappropriate to build lightweight systems that can solve highly constrained subtasks in the near term; such systems have proved useful for many practical tasks. I am, however, advocating spending more of our collective time thinking, talking and writing about the big picture and building integrated, comprehensive systems because this could fundamentally affect our success in solving individual problems posed by natural language.

References

1. *Beale, S.* 1996. Hunter-Gatherer: Applying Constraint Satisfaction, Branch-and-Bound and Solution Synthesis to Natural Language Semantics. Technical Report, MCCS-96-289, Computing Research Lab, New Mexico State Univ.
2. *Beale, S. & McShane, M.* In preparation. *OntoSem* language analysis (available upon request).
3. *Chinchor, N.* 1997. MUC-7 Named Entity Recognition Task Definition. Version 3.5, September 17, 1997. Available at http://www-nlpir.nist.gov/related_projects/muc/proceedings/ne_task.html
4. *de Marneffe, M., MacCartney, B., & Manning, C. D.* 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC 2006*.
5. *Klein, D., & Manning, C. D.* 2003a. Fast exact inference with a factored model for natural language parsing. *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, Cambridge, MA: MIT Press, pp. 3–10.
6. *Klein, D., & Manning, C. D.* 2003b. Accurate unlexicalized parsing. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics—Volume 1* Pages 423–430.
7. *McShane, M.* 1999. The ellipsis of accusative direct objects in Russian, Polish and Czech. *Journal of Slavic Linguistics* 7(1): 45–88.
8. *McShane, M.* 2000a. Hierarchies of parallelism in elliptical Polish structures. *Journal of Slavic Linguistics*, vol. 8, pp. 83–117.
9. *McShane, M.* 2000b. Verbal Ellipsis in Russian, Polish and Czech. *Slavic and East European Journal* 44(2): 195–233.
10. *McShane, M.* 2005. *A Theory of Ellipsis*. Oxford University Press.
11. *McShane, M.* 2009. Reference resolution challenges for an intelligent agent: The need for knowledge. *IEEE Intelligent Systems*, vol. 24, no. 4, pp. 47–58, July/Aug. 2009.
12. *McShane, M.* Submitted. *Toward Automating the Resolution of Difficult Referring Expressions*.
13. *McShane, M., Beale, S. & Nirenburg, S.* 2004. Some meaning procedures of Ontological Semantics. In: Lino MT, Xavier MF, Ferreira F, Costa R, Silva R, editors. *Proceedings of the fourth international conference on language resources and*

- evaluation (LREC-2004). Conference CD distributed by European Language Resources Association (ELRA), Paris, France.
14. *McShane, M., Nirenburg, S., & Beale, S.* 2005. An NLP lexicon as a largely language independent resource. *Machine Translation* 19(2): 139–173.
 15. *McShane, M., Fantry, G., Beale, S., Nirenburg, S., & Jarrell, B.* 2007. Disease interaction in cognitive simulations for medical training. In: Oxley L., Kulasiri D., editors. MODSIM 2007 International congress on modelling and simulation. Modelling and Simulation Society of Australia and New Zealand, 2007. p. 74–80. Virginia Beach, Sept. 11–13, 2007.
 16. *McShane, M., Beale, S., & Nirenburg, S.* 2010. Reference resolution supporting lexical disambiguation. Proceedings of the Fourth IEEE International Conference on Semantic Computing, Carnegie Mellon University, Pittsburgh, PA, Sept. 22–24.
 17. *McShane, M., & Nirenburg, S.* 2012a. A knowledge representation language for natural language processing, simulation and reasoning. *International Journal of Semantic Computing* 6(1).
 18. *McShane, M., Beale, S., Nirenburg, S., Jarrell, B., & Fantry, G.* 2012b. Inconsistency as diagnostic tool in a society of intelligent agents. *Artificial Intelligence in Medicine (AIIM)*, 55(3): 137–48.
 19. *McShane, M., Nirenburg, S., Beale, S., & Johnson, B.* 2012c. Resolving Elided Scopes of Modality in OntoAgent. *Advances in Cognitive Systems*, vol. 2, Dec. 2012.
 20. *McShane, M., Nirenburg, S., Beale, S. & Johnson, B.* 2012d. Resolving Elided Scopes of Modality in OntoAgent. *Advances in Cognitive Systems*, vol. 2.
 21. *McShane, M., Nirenburg, S., Beale, S., Jarrell, B., Fantry, G., & Mallott, D.* 2013a. Mind-, body- and emotion-reading. Proceedings of IACAP 2013 (International Association for Computing and Philosophy), University of Maryland College Park, July 15–17.
 22. *McShane, M., Nirenburg, S., & Jarrell, B.* 2013b. Modeling decision-making biases. *Biologically-Inspired Cognitive Architectures (BICA) Journal*. Volume 3, January: pages 39–50.
 23. *McShane, M., & Nirenburg, S.* 2013c. Use of ontology, lexicon and fact repository for reference resolution in Ontological Semantics. In Oltramari, A., Vossen, P., Qin, L., Hovy, E. (Eds.), *New Trends of Research in Ontologies and Lexical Resources*. Springer. *New Trends of Research in Ontologies and Lexical Resources, Theory and Applications of Natural Language Processing*; pp. 157–185.
 24. *Nirenburg, S., & Raskin, V.* 2004. *Ontological Semantics*. Cambridge, Mass.: The MIT Press.
 25. *Nirenburg, S., McShane, M., & Beale, S.* 2008. A Simulated Physiological/Cognitive “Double Agent”. In: Beal J, Bello P, Cassimatis N, Coen M, Winston P, editors., *Papers from the AAAI fall symposium, naturally inspired cognitive architectures*, Washington, D. C., Nov. 7–9. AAAI technical report FS-08-06, Menlo Park, CA: AAAI Press.
 26. *Tanenhaus M., Spivey-Knowlton M. J., Eberhard K. M., & Sedivy J. C.* 1995. Integration of visual and linguistic information in spoken language comprehension *Science*; Washington; Jun 16, 1995. Volume: 268 Issue: 5217 Start Page: 1632.