7 Ontology, lexicon, and fact repository as leveraged to interpret events of change

Marjorie McShane, Sergei Nirenburg, and Stephen Beale

7.1 Introduction

A semantically insightful way to describe events of change is in terms of their preconditions and effects. For example, if a car accelerates, the value of the 'speed' attribute as applied to the car's motion is higher in its effect than it had been in the precondition of the acceleration event; and if the importance of a certain political theory increases, the value of the modality 'saliency' scoping over that political theory is higher in its effect than it had been in the precondition of the increase event. Consider the following examples of change events drawn from the Wall Street Journal corpus covering 1987, 1988, 1989:

1. Other food stocks rallied in response to the offer for Kraft. Gerber Products **shot up** $4\frac{3}{8}$ to $57\frac{3}{4}$, CPC International **rose** $2\frac{1}{4}$ to $55\frac{1}{2}$, General Mills **gained** $2\frac{1}{8}$ to $54\frac{1}{2}$, Borden **rose** $1\frac{3}{4}$ to $56\frac{1}{2}$, Quaker Oats **went up** $1\frac{3}{8}$ to $56\frac{3}{8}$ and H.J. Heinz **went up** $1\frac{3}{8}$ to $47\frac{3}{4}$.
2. This development follows recent proposals from Saudi Arabia, Kuwait, the United Arab Emirates and Qatar that the current OPEC production ceiling of 16.6 million barrels a day should be **increased** by 900,000 barrels a day.
3. In 1985, 3.9 million women were enrolled in four-year schools. Their number **increased** by 49,000 in 1986.
4. Interco **shot up** 4 to $71\frac{3}{4}$ after a Delaware judge barred its poison pill defense against the Rales group's hostile $74 offer.
5. An index of longterm Treasury bonds compiled by Shearson Lehman Brothers Inc. **rose** by 0.79 point to 1262.85.
6. Stocks of copper in New York's Commodity Exchange warehouses as of the close of business Monday **fell** another 2,250 tons to 52,540 tons. 'Keep your eye on the premium of December over March,' said William O'Neill, research director for Elders Futures Inc. 'The way the premium goes, the market will go.' He said that if the premium continues to **increase**, the market will **go higher**...

For a text processor to glean from such contexts the same information that humans would, it must be able to, among other things:

a. recognize different lexical ways of expressing the same or very similar meanings, like *shot up*, *rose, gained*, *went up* in (1);[1]
b. recognize different syntactic means of expressing the same relationship: the amount by which something rises can be expressed either by a direct object (*rose* $2\frac{1}{4}$ *to* $55\frac{1}{2}$ in (1)) or by a prepositional phrase (*rose by 0.79 point* in (5)); (2) uses the passive voice in *the production ceiling should be increased*, while (6) uses the middle voice in *the premium continues to increase*;
c. recognize phrasal entities, like *shot up* and *went up* in (1), and *go higher* in (6);
d. carry out word-sense disambiguation, both for events and for their associated arguments: the spatial meanings of *shot up* and *rose* in (1), (5) and (6) must be excluded; *premium*, in (6), can have a dozen different meanings, but here it refers to the amount at which something is valued above its par value;
e. calculate a third property value based on two given values: the starting value for each of the stocks mentioned in (1) can be calculated based on the amount of change, the direction of change (increase), and the final value; the final value for the number of barrels in (2) and the number of women in four-year schools in (4) can be calculated based on starting values, amount of change and direction of change (increase);
f. restore elided information: the values provided in (1) and (4) are in dollars per share; the measurement unit of 1262.85 and 0.79 in (5) is points; the premium referred to in (5) is for copper, as implied by the preceding context;
g. understand non-literal language: in (1), *Gerber Products shot up* means that the price per share for the company called Gerber Products shot up (likewise for the other companies listed in (1) and (4)); in (6), *stocks of copper . . . fell* means that the value of stocks of copper fell.

If a system could do all the types of processing listed above, it should be able to perform the same sorts of tasks after reading such texts as a human could, such as answering the following questions:[2]

[1] These can be called near-synonyms, a term borrowed from Graeme Hirst (Hirst, 1995). Within OntoSem, if near-synonyms are not distinguished by property values, it is typically because, for the given domain, we do not yet have sufficient resources to pursue fine semantic distinctions.

[2] Although the current version of the 1980s' corpus from which we drew these examples does not include datelines, we expect most news articles, emails, etc. that we process to include text or metadata indicating the date. Specifying the date, or somehow grounding the question in a given time period, would be expected in a question-answering application.

- What was the price per share of Gerber Products and CPC International before an offer was made to buy Kraft?
- How many barrels of oil a day does Saudi Arabia think OPEC should be able to produce?
- How many women (or 'female students' or 'co-eds') were enrolled in four-year schools in 1986?

Without semantic processing, a system must essentially operate at the level of text strings, extracting only the information that is supplied explicitly, in the local context, and in the expected lexico-syntactic form(s). With semantic processing and a multifaceted knowledge base like the one we will describe, a system can have access to information that is overt or implied, supplied in any of an array of lexico-syntactic forms, and supplied either in a local or non-local context. In this chapter, we seek to show how this latter approach promises significant returns in one semantic realm: events of change.

7.2 A snapshot of OntoSem

OntoSem is a text-processing environment that takes as input raw text and carries out its tokenization, morphological analysis, syntactic analysis, and semantic analysis to yield **text-meaning representations (TMRs)**. Text analysis relies on:

- The OntoSem language-independent **ontology**, which currently contains over 8,000 concepts, amounting to over 140,000 RDF (Resource Description Framework) resource-property-value triples (Java *et al.*, 2006). The meta-vocabulary of the ontology is comprised of about 350 basic properties (relations and attributes). The number of concepts is intentionally restricted, such that a given ontological concept is typically used, with necessary local modifications, in the lexical descriptions of many words and phrases, not only synonyms.
- Entries in an OntoSem **lexicon** for each language processed contain, among other information, syntactic and semantic zones (linked through special variables) as well as procedural-semantic attachments, which we call *meaning procedures*. The semantic zone most frequently invokes ontological concepts, either directly or with modifications, but can also describe word meaning extra-ontologically, for example, in terms of parameterized values of modality, aspect, time, or combinations thereof. The English lexicon currently contains about 35,000 senses, not counting word forms productively analysed on the fly using lexical rules (taking those forms into account significantly increases the number of understood senses in our lexicon).
- A **fact repository**, which is a persistent knowledge base of what logicians call assertions. It contains real-world facts represented as numbered

remembered instances of ontological concepts: e.g., SPEECH-ACT-3186 is the 3,186th instantiation of the concept SPEECH-ACT in the world model constructed during text processing.

- The OntoSem text **analysers**, covering everything from tokenization to extended TMR creation (extended TMRs differ from basic TMRs in that they reflect the results of procedural semantic reasoning).
- The TMR language, which is the **metalanguage** for representing text meaning and is compatible with the metalanguage of the ontology and the fact repository.

The OntoSem environment can actually accomplish, with varying degrees of precision, all the types of processing listed in (a)–(g) of Section 7.1; naturally, lexical coverage is not complete, and cases of ellipsis and non-literal language are especially challenging.[3]

The core of semantic analysis, as we view it, is creating an unambiguous, language-independent representation of text meaning. Our preferred approach to creating applications is to store information extracted from TMRs in the fact repository, then to query the fact repository as needed. Just as the relationships between *types* of objects and events are stored in the ontology (the descriptive component of the knowledge base), the relationships between *instances* of objects and events are stored in the fact repository (the assertion component of the knowledge base). The fact repository, in fact, stores what have recently become known as object and entity profiles. The information in the fact repository both supports the processing of any given text (being a substrate of computer-tractable knowledge) and is supplemented by information from that text. The metalanguage and inventory of properties are the same in the ontology and the fact repository. The knowledge stored in the fact repository is decoupled from the form in which it is expressed in text; it can even be extracted from multilingual sources.

7.3 Motivation for pursuing deep analysis of events of change

A key to successfully populating the fact repository is resolving each expression to its context-free anchor. The most obvious example is finding the real-world referents (not just textual coreferents) for pronouns, but relevant phenomena extend much further. To cite just a few examples:

[3] Although space does not permit a substantive justification for this claim, we can point interested readers to recent publications that provide more details: for example, Nirenburg and Raskin, 2004; Beale *et al.*, 2003; McShane *et al.*, 2005, and the publications are available at the ILIT web site: http://ilit.umbc.edu.

- relative time expressions like *two weeks from Thursday* should, if possible be resolved to a specific date, just as *two hours ago* should be resolved to a specific time;
- approximations like *around 50 pounds* should be resolved to actual ranges;
- calculations, like *She is 3 inches taller than Harry*, should be carried out (how tall *is* she?).

These and other related types of processing are carried out in OntoSem using procedural semantic routines called meaning procedures (McShane *et al.*, 2004). Their results add another layer of knowledge to text-meaning representations. That is, the process of generating TMRs is actually composed of two steps: first, the basic semantic dependency structure is established, including the resolution of syntactic and semantic ambiguity, which yields a basic TMR; then specialized reasoners about language and the world are launched in order to further concretize the TMR. The resulting TMRs – called extended TMRs – show calculated, specific values wherever possible. It is these values that are used to populate the fact repository. The result is that the information in the fact repository is truly divorced from the idiosyncrasies of its source text and from any particular natural language.

Events of change provide an interesting case study for discussing semantic analysis and the use of the fact repository as a query space for applications. At the **ontological level**, events of change need not be listed individually, as is typically done in ontologies and wordnets, but can rather be explained in the lexicon (and, subsequently, in the TMR and fact repository) in terms of their preconditions and effects. At the **lexical level**, the implied information lurking in expressions about events of change must be interpreted differently based on the actual syntactic and semantic input; automating this process requires close lexical specification of relevant lexemes. At the **procedural semantic level**, the actual value of some property before or after an event of change often must be calculated based on the start value or end value in conjunction with the amount and direction of change. At the **fact repository level**, the anchor upon which a calculation must be carried out is often not provided in the local context but, instead, is either an aspect of general knowledge or is supplied elsewhere in the text. In the next section, we will consider *increase* and its synonyms and hyponyms as representatives of events of change. We will describe their lexical and runtime treatment, as well as the TMRs and the fact repository knowledge that they produce.

7.4 *Increase*

Increase and other semantically similar words expressing change are typically recorded in ontologies and wordnets as concepts; after all, they are events, like *run* or *eat*, and could thus be argued to have ontological status. However, are

they *really* like *running* or *eating*? We suggest that they are not, primarily because whenever a verb of change is used, the questions 'which property value changed?', 'in which direction?', and 'by how much?' insinuate themselves into the context. In fact, the comparative values of some property (or properties) in the precondition and effect *constitute* the meaning of verbs of change.

This interpretation of change events has led us to conflate the notion 'event of change' into a single ontological concept, CHANGE-EVENT, which is quite high on the tree of inheritance: its parents are PHYSICAL-EVENT and MENTAL-EVENT, which themselves are children of EVENT. As a high-level ontological concept, CHANGE-EVENT does not have very constrained property values. The main work of interpreting CHANGE-EVENTs, therefore, lies in the further specification of their preconditions and effects. All lexical items that convey change events are mapped to CHANGE-EVENT and are further described in the lexicon using preconditions, effects, and, if needed, other property values: for example, the extent and speed of change is different in *shoot up* than it is in *increase*. This information is housed in the lexicon because the form of lexical input, including the arguments and adjuncts of verbs of change, must be considered when positing an interpretation for any given verb of change in a given context.

In the following subsections, we describe four types of semantic contexts requiring different interpretations of *increase* and its synonyms/hyponyms. Specifically, we consider these verbs as applied to **scalar attributes** (7.4.1), **modalities** (7.4.2), **count nouns** (7.4.3), and **non-count nouns** (7.4.4). For brevity's sake, for each subtype we present just one lexical sense of just one verb (*increase*): for example, we discuss the intransitive *the weight of the elephant increased*, but not the corresponding transitive *they increased the weight of the elephant* or the nominal *the increase in the weight of the elephant*, all of which are treated similarly. All in all, we currently have eighteen senses of *increase*. A summary of the first six is provided in Figure 7.1 for orientation. The other twelve senses include six transitive senses and six nominal senses. For all senses except v4, the synonyms that are treated similarly are *grow*, *escalate*, *rise*, *climb*, and *go up*. The hyponyms that are also treated similarly but include additional semantic features are *soar*, *jump*, *skyrocket*, *shoot up*, *inch up*, and *creep up*: for example, *skyrocket* has the additional features (SPEED (>= .9)), (INTENSITY (>= .9)). Factoring in all the relevant lexemes and syntactic configurations, over 500 lexico-syntactic input combinations are covered by this microtheory of increase events (and a similar number are covered by the corresponding treatment of decrease events).

7.4.1 Increase with scalar attributes

Among the properties defined in the PROPERTY branch of the OntoSem ontology are SCALAR-ATTRIBUTEs, a sample of which includes COMPLEXITY,

Sense	Description of Sense	Example
v1	Intransitive. Subject is a SCALAR-ATTRIBUTE whose RANGE goes up.	The height <the height of the tree, the tree's height> increased ((by) z) (from x) (to y).
v2	Intransitive. Subject is a word expressing an ontological semantic modality whose value goes up.	The importance <the importance of the plan, the plan's importance> increased.
v3	Intransitive. Subject is a count noun whose CARDINALITY goes up.	Mosquitoes <cases of the flu> increased ((by) z) (from x) (to y).
v4	Intransitive. Subject is a non-count noun whose AMOUNT goes up.	Potable water increased ((by) z) (from x) (to y).
v5	Intransitive. Subject is a noun-noun compound in which the second noun denotes an EVENT and the first noun denotes the count-noun THEME of that EVENT, whose CARDINALITY goes up.	Cigarette smoking increased ((by) z) (from x) (to y).
v6	Intransitive. Subject is a noun-noun compound in which the second noun denotes an EVENT and the first noun denotes the non-count-noun THEME of that EVENT, whose amount goes up.	Tofu production increased ((by) z) (from x) (to y).

Figure 7.1 Summary of the intransitive senses of *increase*

COST, INTENSITY, USEFULNESS, RAPIDITY, AGE, and ABSTRACTNESS. While SCALAR-ATTRIBUTEs can take various types of OBJECTs or EVENTs as their domain, they all take a numerical value, or range of values, as their range. That value can either be a real value or a point on the abstract {0,1} scale. For example, *expensive* is lexically described as .8 on the scale of COST. (Of course, one could quibble about whether *expensive* should be rendered as .7, .8, between .7 and .9, etc., but lingering over such questions does not support practical solutions.) Automatic reasoning systems can interpret these abstract values relative to the ontologically listed normal range of property values listed for a concept: for example, an expensive car is a car whose COST is around .8 of the maximum COST listed for cars. During lexicon acquisition we attempt to be consistent in our interpretation of points on the abstract scale: just as *expensive* is .8 on the scale of COST, *tall* is .8 on the scale of HEIGHT and *heavy* is .8 on the scale of WEIGHT. Likewise, *very* consistently shifts the given value by .1 towards the extreme of the scale, and *extremely* shifts it by .2, so *very heavy* will be .9 on the scale of WEIGHT and *extremely heavy* will be 1

```
(increase-v1
 (def  "of scalar attributes: the value of the range is greater in the effect than in
       the precondition of the event")
 (ex   "The weight of the elephant increased ((by) 500 lbs.) (from 1000 lbs.) (to
       1500 lbs.)")
 (syn-struc
   ((subject ((root $var1) (cat n)))
    (root $var0) (cat v)
    (pp ((root $var2) (cat prep) (root by) (prep-head-opt +) (opt +)
        (obj ((root $var3) (cat n)))))
    (pp ((root $var4) (cat prep) (root from) (opt +)
         (obj ((root $var5) (cat n)))))
    (pp ((root $var6) (cat prep) (root to) (opt +)
        (obj ((root $var7) (cat n)))))))
 (sem-struc
   (CHANGE-EVENT
    (PRECONDITION (value refsem1))
    (EFFECT (value refsem2))
    (CHANGE-IN-VALUE ((value ^$var3) add)))
   (refsem1 (SCALAR-ATTRIBUTE  (RANGE (value ^$var5))))
   (refsem2 (SCALAR-ATTRIBUTE (RANGE (value ^$var7))))
   (< (value refsem1.RANGE) (value refsem2.RANGE))
   (^$var2 (null-sem +))  (^$var4 (null-sem +)) (^$var6 (null-sem +)))
 (meaning-procedure
   (seek-specification (value refsem1) (value ^$var1))
   (seek-specification (value refsem2) (value ^$var1))
   (fill-in-missing-values-for-increase-v1)))
```

Figure 7.2 The first verbal sense of *increase*

(see McShane *et al.*, 2004 for further discussion of modifications to scalars). For change events, it is the relative values of scalar attributes that are most important: if the size of something increases, the value of its range is higher in the effect of the change event than in its precondition. Let us consider how one of the verbal senses for *increase* (*increase-v1* in Figure 7.2) supports such an interpretation, after some background is provided on the formalism used in OntoSem.

Lexicon entries are written using an extended variety of LFG in LISP-compatible format. Elements of the syntactic structure (*syn-struc*) and semantic structure (*sem-struc*) are linked using variables, and the same variables are referred to in the *meaning-procedure* zone. The caret (^) preceding variable names in the sem-struc indicates 'the meaning of (the variable)'. *Refsem*

is a reserved term used for certain coreference needs within an entry. Ontological concepts are written in SMALL CAPS. The optionality of categories is indicated by *(opt +)*, noted at the level of the head of the category. The feature *(prep-head-opt +)* indicates that if the PP is overt, its preposition may be elided (to account for the difference between *The cost increased $10* and *The cost increased by $10*). The descriptor *(null-sem +)* indicates that no compositional meaning should be attributed to the prepositions since their meaning – or, more precisely, function – has already been taken care of in the semantic description.

Let us start by considering a sentence in which none of the optional PPs is overt in the context, such as *The weight of the elephant increased.* The meaning of *increase* can be rendered, 'the value of the range of WEIGHT is greater in the effect than in the precondition of *increase*'. This information is captured in the sem-struc statement:

(<(value refsem1.RANGE) (value refsem2.RANGE))

Refsem1 and refsem2 refer to some SCALAR-ATTRIBUTE, whose specific meaning can only be understood after the subject of the clause has been semantically analysed. The procedural semantic routine that seeks the meaning of certain elements of input and uses that meaning to interpret other elements of input is called *seek-specification.* The call to this routine, which is listed in the *meaning-procedure* zone of the entry, essentially says 'Seek the meaning of the SCALAR-ATTRIBUTE referred to by refsem1 and refsem2 using the meaning of $var1 as an input parameter.' Assuming that the subject of the sentence is *the weight of the elephant*, as in our example, the analyser will select the sense of the word *weight* in Figure 7.3.[4]

This sense indicates that the word *weight* instantiates the ontological concept WEIGHT whose DOMAIN is the meaning of $var2, which in our example instantiates the concept ELEPHANT. The result of the *seek-specification* meaning procedure, therefore, is replacement of the SCALAR-ATTRIBUTE instantiated by *increase* with WEIGHT (DOMAIN ELEPHANT) in the TMR. The TMR

[4] One bit of lexical complexity is worth mentioning. Some words that map to SCALAR-ATTRIBUTEs have two different interpretations: one in which the range of the attribute is unspecified, and the other in which the range is understood as having a high value. For example, height can mean either 'distance from the base of something to the top' or 'the condition or attribute of being relatively or sufficiently high or tall'. Under the first interpretation, the range of HEIGHT is unspecified, whereas under the second interpretation it is, say, (> .7). If one says *I was surprised by the height of that steeple*, the interpretation is (HEIGHT (> .7)), whereas if one says *The height of the tree increased*, the initial and ending HEIGHTs remains unspecified. We are working on developing heuristics for such disambiguation in the context of our larger work on semantic disambiguation (see, e.g., Beale *et al.*, 2003 for an overview of disambiguation in OntoSem).

```
(weight-n1
  (def "indicates physical heaviness")
  (ex "the weight (of the child)")
  (syn-struc
    ((root $var0) (cat n)
     (pp ((root $var1) (cat prep) (root of) (opt +)
          (obj ((root $var2) (cat n)))))))
  (sem-struc
    (WEIGHT
      (DOMAIN (value ^$var2)))
    (^$var1 (null-sem +))))
```

Figure 7.3 The first nominal sense of *weight*

resulting from the input sentence *The weight of the elephant increased* will be as shown in 7.4.[5]

This text-meaning representation was generated from input in which none of the optional PPs for *increase* were overt: that is, there was no indication of the elephant's starting weight, ending weight or change in weight. However, as shown in the 'example' field for *increase-v1*, any combinations of PPs can be overt: *The weight of the elephant increased ((by) 500 lbs.) (from 1,000 lbs.) (to 1,500 lbs.)*. When such optional information is present, it is used in two ways. First, it directly fills slots in the *sem-struc* (e.g., the value of a *to*-PP fills in the RANGE of refsem2), which is then rendered as a more information-rich TMR. Second, if at least two of the three values are provided or can be recovered from the context, the full template of values (the value before the increase, after the increase, and the amount of increase) can be filled in. This is the ideal situation for one of the main applications of OntoSem: populating the fact repository with real-world facts, both explicit in the text and inferred with a high degree of confidence.

Consider the TMR for the input *The weight of the elephant increased by 500 lbs. to 1,500 lbs.* in Figure 7.5. Since we have two values (end value and amount of change), we can calculate the starting value (the RANGE of WEIGHT-3). The meaning procedure called to do this is descriptively named *fill-in-missing-values-for-increase1*. It is actually a multi-part routine (whose full call is not presented here) that does several things: if two values are present,

[5] '(< find-anchor-time)' is a call to a meaning procedure that seeks the anchor time in a dateline or other text source. Since the anchor time cannot be resolved in our short context, the reference to the meaning procedure (which means 'prior to the anchor time' and reflects the past tense of the verb) remains in the TMR.

CHANGE-EVENT-1

textpointer	increased
PRECONDITION	WEIGHT-1
EFFECT	WEIGHT-2
TIME	(< find-anchor-time)
(< WEIGHT-1.RANGE	WEIGHT-2.RANGE)

WEIGHT-1

textpointer	weight
DOMAIN	ELEPHANT-1
PRECONDITION-OF	CHANGE-EVENT-1

WEIGHT-2

textpointer	weight
DOMAIN	ELEPHANT-1
EFFECT-OF	CHANGE-EVENT-1

ELEPHANT-1

textpointer	elephant
DOMAIN-OF	WEIGHT-1 WEIGHT-2

Figure 7.4 The TMR for *The weight of the elephant increased*

it calculates the third; if only the amount of change is present, it seeks out the initial or final value from the preceding context (more specifically, the TMR for the preceding context) and, if successful, uses the now-known two values to calculate the third one. Notice that the *fill-in-missing-values-for-increase1* has a different status than *seek-specification*: whereas *seek-specification* is essential to interpreting the actual textual input, *fill-in-missing-values-for-increase1* attempts to go beyond the input in order to arrive at a fuller representation of all available meaning for the fact repository. Although we have been talking about the example of weight throughout, we must emphasize that this sense of *increase* covers input in which the subject refers to any scalar attribute.

One final note is worth mentioning before we leave the topic of scalar attributes. *Increase* is a particularly complex example since one cannot record beforehand which scalar attribute is in question: that must be compositionally determined. However, for many other lexemes, the relevant scalar attribute can be lexically encoded: e.g., *to accelerate* refers to increasing the value of VELOCITY, *to smooth out* refers to increasing the value of SMOOTHNESS, and *to dry* refers to decreasing the value of WETNESS. The description of such lexemes and their representation in TMR are very similar to the case of *increase* except that the *seek-specification* procedural routine is not required.

CHANGE-EVENT-2

textpointer	increased
PRECONDITION	WEIGHT-3
EFFECT	WEIGHT-4
CHANGE-IN-VALUE	+ 500 (MEASURED-IN POUND)
TIME	(< find-anchor-time)

(< WEIGHT-3.RANGE WEIGHT-4.RANGE)

WEIGHT-3

textpointer	weight
DOMAIN	ELEPHANT-2
PRECONDITION-OF	CHANGE-EVENT-2

WEIGHT-4

textpointer	weight
DOMAIN	ELEPHANT-2
RANGE	1500 (MEASURED-IN POUND)
EFFECT-OF	CHANGE-EVENT-2

ELEPHANT-2

textpointer	elephant
DOMAIN-OF	WEIGHT-3 WEIGHT-4

Figure 7.5 The TMR for *The weight of the elephant increased by 500 lbs. to 1,500 lbs.*

7.4.2 Increase with modalities

Modalities express an attitude on the part of the speaker towards the content of a proposition.[6] Within OntoSem, they are treated as extra-ontological aspects of meaning. The modalities currently used in OntoSem are: *epistemic*, *belief*, *obligative*, *permissive*, *potential*, *evaluative*, *intentional*, *epiteuctic*, *effort*, *volitive*, and *saliency*. The scale for the values of each modality is {0,1}, with any decimal value or range in between being valid. Examples of lexical items whose meanings are conveyed by values of modality are: *must* – obligative 1; *might* – epistemic .4 <> .6; *important* – saliency .8; *loathe* – evaluative 0. Modalities are defined for *type*, *scope*, *value*, and *attributed-to*, with the latter defaulting to the speaker if not overtly specified. For example, one sense of *importance* is described as in Figure 7.6.[7]

[6] There is often semantic ellipsis of part of the proposition: e.g., if one says that honour is important, the proposition is a person's having the attribute of being honourable.

[7] Another sense of *importance* is 'high level of importance', in which the value for saliency would be (> .7). Cf. footnote 4.

```
(importance-n1
 (cat n)
 (def "some unspecified value of the modality 'saliency'")
 (ex "the importance of taking vitamins")
 (comments "This sense is used primarily in contexts of comparison and events
   of change; the sense 'highly important' is the default otherwise.")
 (syn-struc
  ((root $var0) (cat n)
   (pp ((root $var1) (cat prep) (root of) (opt +)
        (obj ((root $var2) (cat n)))))))
 (sem-struc
  (modality
    (type saliency)
    (scope (value ^$var2)))
  (^$var1 (null-sem +))))
```

Figure 7.6 The first nominal sense of *importance*

Values for modality are compared in inputs like the following, extracted from a corpus: *The importance of this issue ⟨ The need for help, The emphasis on health care, Efforts by insurers ⟩ increased.* The lexical sense of *increase* that covers such inputs is shown in Figure 7.7.

This sense assumes, as appears to be justified from a small corpus study, that PP adjuncts indicating exact values will not typically be used with modals (modifiers like *a lot* and *significantly* will be treated compositionally and need not be referred to in the entry for *increase*).[8] Apart from its relative simplicity due to a lack of optional PPs, *increase-v2* is actually quite similar to *increase-v1*, the differences reducing to those listed in Figure 7.8. The analysis of an input like *The importance of diplomacy increased* will produce the TMR as in Figure 7.9.

The semantic analyser can disambiguate between *increase-v1* and *increase-v2* because each sense imposes semantic constraints on ^$var1: in *increase-v1*, it must be a SCALAR-ATTRIBUTE and, in *increase-v2*, it must be a type of modality.

7.4.3 Increase with count nouns

Ellipsis in language is widespread, which poses well-known problems for NLP (McShane, 2005). However, some cases of ellipsis are predictable and can be

[8] If a PP that indicates the starting value, ending value or amount of change should happen to be used with *increase-v2*, its meaning can be arrived at compositionally, though disambiguation of the preposition(s) in question will need to be carried out. The main reason for listing optional PPs explicitly in certain senses of *increase* is to circumvent the need for run-time disambiguation of prepositions.

```
(increase-v2
 (def "used with modalities: the value of the modality increases")
 (ex "The importance of diplomacy has increased.")
 (syn-struc
   ((subject ((root $var1) (cat n)))
    (root $var0) (cat v)))
 (sem-struc
   (CHANGE-EVENT
     (PRECONDITION (value refsem1))
     (EFFECT (value refsem2)))
   (refsem1 (modality))
   (refsem2 (modality))
   (< (value refsem1.value) (value refsem2.value)))
 (meaning-procedure
   (seek-specification (value refsem1) (value ^$var1))
   (seek-specification (value refsem2) (value ^$var1))))
```

Figure 7.7 The second verbal sense of *increase*

	increase-v1	**increase-v2**
Which sem-struc element requires procedural-semantic specification?	SCALAR-ATTRIBUTE	The type of modality
What changes from the precondition to the effect of the CHANGE-EVENT?	The range of the SCALAR-ATTRIBUTE	The value of the modality

Figure 7.8 Comparing *increase-v1* and *increase-v2*

handled using a combination of static resources and programs that use them. Consider the following corpus excerpts:

- After that **the mosquitoes increased** and there was a considerable amount of fever in October and November.
- Following cessation of wolf control in 1960 **wolves increased** and attained densities of approximately 16 wolves/1000 km^2 by 1970.
- As Figure 2 shows, **total accidents increased** modestly from 1993 through 1997.

CHANGE-EVENT-3

textpointer	increased
PRECONDITION	modality-1
EFFECT	modality-2
TIME	(< find-anchor-time)
(< modality-1.value	modality-2.value)

modality-1

textpointer	importance
SCOPE	DIPLOMATIC-EVENT-1
TYPE	saliency
PRECONDITION-OF	CHANGE-EVENT-3

modality-2

textpointer	importance
SCOPE	DIPLOMATIC-EVENT-1
TYPE	saliency
EFFECT-OF	CHANGE-EVENT-3

DIPLOMATIC-EVENT-1

textpointer	diplomacy
scope-of	modality-1 modality-2

Figure 7.9 The TMR for *The importance of diplomacy increased*

The implications of the above three sentences are, respectively, that the number of mosquitoes, the number of wolves, and the number of accidents increased, even though there is no explicit reference to *number* in any of the contexts. The pivotal clue that underpins the automated interpretation of such contexts is the recognition that the subject NP is a count noun – regardless of whether it refers to an ontological OBJECT (MOSQUITO, WOLF) or EVENT (ACCIDENT).

In OntoSem, count and non-count are not defined lexically, they are defined ontologically. Count nouns are mapped to concepts which are in the domain of CARDINALITY, whereas non-count nouns are mapped to concepts which are in the domain of AMOUNT. Roughly speaking, MATERIAL and INTANGIBLE-OBJECT are defined for AMOUNT, whereas all other OBJECTs and EVENTs are defined for CARDINALITY. The semantic analyser will select *increase-v3* (Figure 7.10) only in those contexts in which the subject maps to an entity whose ontological mapping is in the domain of CARDINALITY.

Note that no meaning procedure is required to seek the specification of the property in question: the property CARDINALITY is asserted to be the

```
(increase-v3
 (def "used with count nouns")
 (ex "The mosquitoes increased")
 (syn-struc
   ((subject ((root $var1) (cat n)))
    (root $var0) (cat v)
    (pp ((root $var2) (cat prep) (root by) (prep-head-opt +) (opt +)
         (obj ((root $var3) (cat n)))))
    (pp ((root $var4) (cat prep) (root from) (opt +)
         (obj ((root $var5) (cat n)))))
    (pp ((root $var6) (cat prep) (root to) (opt +)
         (obj ((root $var7) (cat n)))))))
 (sem-struc
   (CHANGE-EVENT
     (PRECONDITION (value refsem1))
     (EFFECT (value refsem2))
     (CHANGE-IN-VALUE ((value ^$var3) add)))
   (refsem1
     (set
       (element-type (value ^$var1))
       (CARDINALITY (value ^$var5))))
   (refsem2
     (set
       (element-type (value ^$var1))
       (CARDINALITY (value ^$var7))))
    (< (value refsem1.CARDINALITY) (value refsem2.CARDINALITY))
    (^$var2 (null-sem +))  (^$var4 (null-sem +)) (^$var6 (null-sem +)))
 (meaning-procedure (fill-in-missing-values-for-increase-v3)))
```

Figure 7.10 The third verbal sense of *increase*

one in question for all inputs whose ^$var1 refers to a count OBJECT or EVENT. The TMR produced for the input *The mosquitoes increased* is shown in Figure 7.11.

7.4.4 Increase with non-count nouns in N-N compounds

A similar type of semantic ellipsis can occur with non-count nouns: an *amount* of something can be referred to without the word *amount*, as in *Potable water increased*. For the sake of variety, let us consider a lexical sense of increase that treats implied amounts but in a slightly more complex syntactic structure – one

CHANGE-EVENT- 4
textpointer increased
PRECONDITION SET-1
EFFECT SET-2
TIME (< find-anchor-time)
(< set-1.CARDINALITY set-2.CARDINALITY)
set-1
ELEMENT-TYPE MOSQUITO-1
PRECONDITION-OF CHANGE-EVENT-4
set-2
ELEMENT-TYPE MOSQUITO-1
EFFECT-OF CHANGE-EVENT-4
MOSQUITO-1
textpointer mosquitoes
CARDINALITY (> 1)
element-of set-1 set-2

Figure 7.11 The TMR for *The mosquitoes increased*

in which the subject is a noun-noun compound. The configuration in question is illustrated by examples like the following: *Wine consumption ⟨ Calcium intake, Cocaine use ⟩ increased*. Here, the first noun in each N-N compound has two notable properties: it is the THEME of the EVENT referred to by the second noun of the compound, and its AMOUNT is understood to increase. The lexical sense that covers such contexts is *increase-v6*, as shown in Figure 7.12.

The *syn-struc* of this entry explicitly requires an N-N compound as its subject. The *sem-struc* says that there is a CHANGE-EVENT, through which the AMOUNT of ^$var1 (WINE) in the PRECONDITION is less than in the EFFECT. The *sem-struc* also asserts that ^$var1 (WINE) is the THEME of ^$var2 (DRINK), and that ^$var2 itself must be an EVENT. This latter semantic constraint supports disambiguation between *Wine consumption increased (increase-v6)* and *Wine vinegar increased*. The latter would be covered by *increase-v4*, a sense, not shown here, that expects the subject to be a non-count entity.

Increase-v6 as applied to the input *Wine consumption increased* yields the TMR shown in Figure 7.13.

7.5 Content divorced from its rendering

One of the main emphases of OntoSem text processing is separating content from form. For practical purposes, *The price increased by $2.00 to $22.00* and *The price increased by $2.00 from a starting price of $20.00* are synonymous,

```
(increase-v6
 (def "intransitive; the subject is a N-N compound in which the first N is a non-
   count noun")
 (ex "Wine consumption increased ((by) 10%) (from 10 glasses per month) (to
   11 glasses per month)")
 (syn-struc
   ((subject
      (n ((root $var1) (cat n)))
      (n ((root $var8) (cat n))))
    (root $var0) (cat v)
    (pp ((root $var2) (cat prep) (root by) (prep-head-opt +) (opt +)
        (obj ((root $var3) (cat n)))))
    (pp ((root $var4) (cat prep) (root from) (opt +)
        (obj ((root $var5) (cat n)))))
    (pp ((root $var6) (cat prep) (root to) (opt +)
        (obj ((root $var7) (cat n)))))))
 (sem-struc
   (CHANGE-EVENT
     (PRECONDITION (value refsem1))
     (EFFECT (value refsem2))
     (CHANGE-IN-VALUE ((value ^$var3) add)))
   (^$var8 (sem EVENT)
      (THEME (value ^$VAR1)))
   (refsem1
     (AMOUNT
        (DOMAIN (value ^$var1))
        (RANGE (value ^$var5))))
   (refsem2
     (AMOUNT
        (DOMAIN (value ^$var1))
        (RANGE (value ^$var7))))
   (< (value refsem1.RANGE) (value  refsem2.RANGE))
   (^$var2 (null-sem +))  (^$var4 (null-sem +)) (^$var6 (null-sem +)))
   (meaning-procedure (fill-in-missing-values-for-increase-v6)))
```

Figure 7.12 The sixth verbal sense of *increase*

and a human reader/listener who recalls the information later will likely not remember in which form it was presented. Similarly, we would want reasoning-oriented NLP systems to extract and record the same information from two synonymous texts regardless of the form in which the information was expressed in each source. In the example just cited, the analysis requirement is calculating the missing value from the 'start-change-end' value

CHANGE-EVENT-5

textpointer	increased
PRECONDITION	AMOUNT-1
EFFECT	AMOUNT-2
TIME	(< find-anchor-time)
(< AMOUT-1.RANGE AMOUNT-2.RANGE)	

AMOUNT-1

DOMAIN	WINE-1
PRECONDITION-OF	CHANGE-EVENT-5

AMOUNT-2

DOMAIN	WINE-1
EFFECT-OF	CHANGE-EVENT-5

WINE-1

textpointer	wine
DOMAIN-OF	AMOUNT-1 AMOUNT-2
THEME-OF	DRINK-1

DRINK-1

textpointer	consumption
THEME	WINE-1

Figure 7.13 The TMR for *Wine consumption increased*

triple – all of whose values are equally accessible to any human interpreter of such a text.[9]

Another requirement of a sufficient NLP system is keeping a record of crucial bits of information presented elsewhere in a communication. A human reader/listener is expected to keep track of non-local information and, over the course of an article, description, etc., construct a unified mental model from its facts. Any approach to NLP that is strictly sentence-based, clause-based or based on the physical proximity of information is bound to miss crucial connections.

A third requirement of a truly intelligent NLP system is making use of what is understood to be general world knowledge. For example, if a newspaper reports that a presidential candidate in some country is even younger than John F. Kennedy was when he ran for president, the reader would be expected to know that JFK was in his early forties when he was elected president of the United States (or, more broadly, to understand that he was young for a president).

[9] For a discussion of paraphrase at the level of TMR, see McShane *et al.*, 2005.

Yet another challenge for an optimal, reasoning-oriented NLP system is to extract information from multilingual sources and merge it into unified knowledge structures. Such a multilingual application – unlike, for example, machine translation – cannot pass ambiguity, underspecification, ellipsis, etc., on to the end user to resolve.

Finally, an intelligent NLP system should be able to carry out reasoning beyond that needed for the interpretation of textual input. It is to such reasoning that we now turn.

7.6 NLP with reasoning and for reasoning

At present, the quality of automatic reasoning for real-world problems is insufficiently high and its coverage insufficiently broad at the level of both system components and knowledge elements. A central contributing factor is brittleness due to (a) the insufficiency of factual, heuristic, and other necessary types of knowledge in current reasoners, (b) a relatively narrow inventory of reasoning methods (deduction is still the main reasoning tool of choice), and (c) the need to find a balance between completeness and soundness of reasoning systems, on the one hand, and their efficiency and utility, on the other.

We have already shown that reasoning is needed for the semantic analysis of text and the piecing together of knowledge elements both from different portions of texts and from different texts altogether. Once a set of such structures is created and stored, it can support further reasoning, since reasoning is almost universally understood as operations that take structured data as input and generate other structured data (see, for example, the JTP system (Fikes *et al.*, 2003)). Without the availability of structured data, like the fact repository we have been discussing, the reasoning enterprise would not be able to relate to the real world. Moreover, in a realistic application, the data must be ample and dynamic, meaning that its knowledge resources must be constantly and promptly augmented. If knowledge is generated largely by people, this latter condition makes it more difficult to expect utility in real-world applications. The knowledge-generation process must be automated, which is why we are pursuing 'smart' automatic augmentation of the fact repository from TMRs.

The areas of general reasoning and NLP are often, though clearly not always, separated in the reasoning community. That is, the history of automated reasoning has shown an unfortunate bifurcation: the separation of language understanding from other aspects of reasoning, which predates even the inception of AI. One of the first acute observations on this split was made by Yehoshua Bar Hillel in the 1950s:

> The evaluation of arguments presented in a natural language should have been one of the major worries of logic since its beginnings. However ... the actual development of

formal logic took a different course. It seems that ... the almost general attitude of all formal logicians was to regard such an evaluation process as a two-stage affair. In the first stage, the original language formulation had to be rephrased, without loss, in a normalized idiom, while in the second stage, these normalized formulations would be put through the grindstone of the formal logic evaluator ... Without substantial progress in the first stage even the incredible progress made by mathematical logic in our time will not help us much in solving our total problem (Hillel, 1970, pp. 202–3).

Once one substitutes 'knowledge representation language' for 'normalized idiom' and 'reasoner' for 'formal logic evaluator', we can see that the current state of affairs is quite similar to that of almost half a century ago.

There are many reasons why such an extension of the purview of reasoning has not really occurred, an important one being that the task of extracting text meaning has been considered too complicated to succeed. However, just as in the area of reasoning, after a long period of intensive theoretical work, many people in NLP have come to realize that application-oriented work requires various kinds of simplification, coarsening of the grain size of description and inferencing, and concentration on the main bulk of knowledge. Theory-oriented work, by contrast, justifiably ponders over difficult cases, paradoxes, and counterexamples.

Both deductive and abductive logical approaches have been applied to NLP. Deductive approaches tend to focus on small, well-defined phenomena – often just one or two rules – and generally do not support the broad-coverage extraction and manipulation of meaning in practical systems (a typical recent contribution is Condoravdi *et al.*, 2003). Abductive approaches, by contrast, have been applied to larger, real-world tasks: typically, an abductive model-based reasoner is supported by (1) an ontology featuring co-occurrence properties (e.g., agents, themes, or instruments of events), (2) knowledge about typical sequences of events (scripts), and (3) the agents' goals and plans (Hobbs *et al.*, 1990, 1997). The grain size of such knowledge is coarser than that used in a typical lexical database reasoner and the results are seldom provably correct, but they represent the most probable conclusion that can be drawn based on the available data.

7.7 Conclusion

We began this discussion with a set of examples that involved events of change – specifically, events conveying increases and decreases of property values. We showed that a variety of lexical and syntactic means could be used to convey essentially the same information, and described the OntoSem methods of capturing such meaning using a language-independent metalanguage grounded in an ontology. We then turned to larger issues of reasoning, including the almost universal need for the input of reasoning systems to be

statements in an unambiguous metalanguage. The NLP community has not yet developed methods of providing this kind of input large-scale. Using OntoSem, we are working towards that goal, interpreting the fact repository both as a target of structured knowledge from NLP and as a source of structured knowledge to support better NLP. Supporting reasoning capabilities is one of the central reasons for pursuing deeper semantic analysis than might otherwise be deemed necessary. Some of these reasoning capabilities must, in fact, be used in the semantic analysis process itself.

7.7.1 Comparing OntoSem resources with others

The chapters in this volume reflect two currently dominant currents in NLP: (1) building resources is a method- and resource-driven undertaking and (2) applications are developed to leverage available resources rather than resources being developed to serve the needs of independently acknowledged applications. We comment on both of these as a means of orienting OntoSem with respect to outside resources and systems.

Most work on developing knowledge resources for NLP over the past ten years has involved automatic methods and already existing resources. Projects have included learning ontologies or wordnets from corpora (e.g., Aramaki *et al.*, 2005), automatically merging ontologies or wordnets (e.g., Chapter 10, Pustejovsky *et al.*, 2002), and generating wordnets in one language by bootstrapping from another language. Such approaches typically have two goals: the first (and in many cases foremost) goal is to develop methods – hone algorithms, test machine-learning engines, etc.; the second is to create large knowledge bases to be used in applications. Although the benefits of such approaches are much discussed, not so the drawbacks: for example, the quality and depth of knowledge offered by the base resources is, across the board, not sufficient to support truly sophisticated applications, and automatic processes launched on those resources – as through merging – only aggravate this problem. In sum, for short-term applications and for long-term applications in which either the depth of semantic analysis is not crucial or errors do not carry a high cost, the resources thus generated are appropriate. However, this description does not cover all applications.

The OntoSem group, by contrast, has a different sphere of interests and, accordingly, different methods for building resources. We are pursuing the rigorous semantic interpretation of language to support high-end applications, with special areas of interest being disambiguation, the detection and resolution of elided and underspecified structures, and reasoning about language and the world. As such, the quality of our static resources needs to be very high, and the only currently feasible method for acquiring them is manually (we

are currently experimenting with bootstrapping our current resources using machine-learning techniques and expect to report results shortly).

An example of a current OntoSem application is medical simulation and tutoring, for which the OntoSem ontology must support both the interactive simulation and the natural language dialogue that permits intelligent agents (virtual patients, virtual diagnosticians, and the virtual tutor) to communicate with human users. Related ontology expansion involves not only recording dozens of property values for each object and event (using multivalued selectional restrictions in the ontology), but also extensive domain and workflow scripts. When we began expanding our general-purpose ontology into the medical domain, we attempted to exploit the Unified Medical Language System (UMLS) metathesaurus and semantic network (see Nirenburg *et al.*, 2005 and cf. Pustejovsky *et al.*, 2002) but soon abandoned the effort because the results we were able to present to physicians for their validation were too noisy (recall that UMLS was created by librarians for librarians).

This is not to say that we do not use outside resources – we do, but to inform rather than displace the manual acquisition process. For example, the lexicon acquisition process regularly involves checking WordNet and dictionary.com for synonyms and hyponyms, and the anatomical aspect of our medical ontology building has been facilitated by the University of Washington's Foundational Model of Anatomy.[10] We have found that pruning or cleaning a noisy resource is no less work than building a resource from scratch, and our acquisition methodolgy reflects this experience.

We have reported elsewhere on the benefits of creating resources to serve applications rather than the other way around (McShane *et al.*, 2004; Nirenburg *et al.*, 2004). However, even if one must develop an application using extant resources, it is essential to understand what those resources actually provide. An often misinterpreted resource, it seems, is WordNet (and its progeny). Developers of WordNet do not oversell its content: 'WordNet is often called an ontology, although its creators did not have in mind a philosophical construct. WordNet merely represents an attempt to map the English lexicon into a network by means of a few semantic relations ... A full semantic inventory of language is beyond what has currently been attempted here. The authors are aware that a much richer corpus is needed ... which would capture a logical semantics for complex template linguistic expressions, rather than individual lexical items' as described in Chapter 2. In short, WordNet is not propounded to be the ultimate NLP lexicon: it is a useful, available tool for some types of applications. However, it seems to be misinterpreted by many as the last word in lexicons for NLP. A case in point is Chapter 10, which discusses methods

[10] Available at http://sig.biostr.washington.edu/projects/fm/FME/index.html

of merging ontologies and lexical resources. Although that survey includes several types of ontologies, the only types of lexical resources considered are those having the structure and content of a wordnet, the reason being that 'nowadays in the literature "WordNet" is the de facto standard for interfacing' (Chapter 10). The key word here is 'nowadays'. We would suggest the need for the community to consider not only the present and very near future when formulating approaches to developing and using resources for NLP: the longer term goal of developing truly intelligent agents is no less compelling a window of opportunity.