# Some Meaning Procedures of Ontological Semantics

**Marjorie McShane, Stephen Beale and Sergei Nirenburg**

Institute of Language and Information Technologies
University of Maryland Baltimore County
{marge, sbeale, sergei}@umbc.edu

**Abstract**

This paper presents implemented algorithms for interpreting the meaning of certain context-dependent lexical items within the Ontological Semantic text processing environment. We discuss the form, function and rationale behind three meaning procedures, all of which are, in a certain sense, numerically oriented. We show that only a knowledge-rich processing system can fully interpret such entities, and that an integrated combination of static resources and processors provides sufficient foundation for high-quality text interpretation.

## Introduction

The meaning of text elements and combinations thereof depends to varying extents on the context in which they appear. Whereas outside of context the noun *sheep* and the verb *dance* conjure a relatively stable image, the adverbials *approximately, very* and *nearly* depend crucially on their context to concretize their meaning. In the Ontological Semantic (OntoSem) text processing environment, the meaning of context-dependent lexical elements is arrived at by a combination of static lexical descriptions (which link to a language-independent ontology) and procedures that attempt to specify text meaning based on ontological, contextual and other information. These *meaning procedures* (MPs) are the topic of this paper.

OntoSem is a text-processing system that supports end-to-end applications, like knowledge extraction, machine translation, and question answering (see, e.g., Nirenburg and Raskin, forthcoming and Nirenburg et al. 2003). The system takes unrestricted text as input, carries out tokenization, morphological analysis, syntactic analysis and semantic analysis, and produces text-meaning representations (TMRs) that are the basis for further applications. It relies on a language-independent ontology, a hand-crafted ontological lexicon , a fact repository (which contains real-world instances of ontological concepts and their properties), an onomasticon (a lexicon of named entities), and a suite of processors. A very simple example of a TMR, reflecting the meaning of the sentence *The US won the war*, is as follows:

WIN-3
    AGENT NATION-213
    THEME WAR-ACTIVITY-7

This TMR is headed by a WIN event – in fact, it is the 3rd instantiation of the concept WIN (WIN-3) in the world model being built during the processing of the given text(s). Its agent is NATION-213, which is the key for US in our fact repository. The theme of the event is the 7th instantiation of WAR-ACTIVITY in this analyzer run.

This TMR is what we call a *basic TMR*, since it reflects basic semantic dependency building, including the resolution of syntactic and semantic ambiguity. There is, however, another level of processing, during which specialized reasoners about language and the world are launched in order to further concretize the TMR. The resulting TMRs – called *extended TMRs* – show the calculated values of various modalities, aspect, time, reference resolution, speaker attitudes, etc. So, extensions to the above simple TMR would: a) include as specific a value for time as possible (i.e., the time of speech must be extracted from the text so that the past-tense verb can be interpreted), and b) attempt to link WAR-ACTIVITY-7 to the appropriate coreferential WAR-ACTIVITY in the fact repository (since the text contains ***the*** *war*, with a definite article, we know that there must be some coreferential war either in the preceding context or available as an aspect of general world knowledge, which should be stored in our fact repository). The results of reasoning are error prone (e.g., programs to resolve pronominal reference will likely never be 100% accurate); as such, extended-TMR elements derived by reasoning are understood as being defeasible.

Since the reasoners can be difficult to develop and error prone, one might ask why we bother to pursue extended TMRs in the first place. While the rationale for carrying out, e.g., reference resolution can be assumed self-evident, the rationale for seeking actual values for expressions like *around 6 p.m.* or *nearly 50 dollars* might not be as clear – especially since it would be difficult to reach widespread consensus regarding what, precisely, *around 6 p.m.* means: is it +/- 5 minutes? 10 minutes? the range between 5 and 15 minutes? The rationale for calculating actual values is to support reasoning: e.g., if one news report says there was an explosion in Jerusalem at *around 5 p.m.* on some date, and another report says there was an explosion in Jerusalem at *4:48 p.m.* on the same date, the coreference relationship between them can be established if *around 5 p.m.* is expanded to *(4:50 <> 5:10 p.m.).* We currently can, in fact, automatically establish such coreference relations as part of the reference resolution procedures in OntoSem. Thus, we are not claiming that the values that we calculate to convey the meaning of words like *about, approximately, nearly*, etc., are the only

or even best possible ones (domain-specific testing will be the judge of that); we are, however, claiming that some reasonable approach to resolving such references will support reasoning better than no such resolution at all.

This paper is devoted to numerically-oriented MPs, a subset of all the MPs in OntoSem. Specifically, we discuss the MPs called *delimit-scale, decrease-/increase-value,* and *specify-approximation.* All of these MPs are associated with specific lexical descriptions, reflecting the expectation-oriented nature of resource acquisition in OntoSem. For example, we know that *approximately* and *nearly* require context-based resolution, so we encode a procedural attachment for them in the MP zone of the respective lexicon entries.

In the next section we present background about the treatment of scalar attributes in OntoSem, which are important for the MPs in question; in fact, this is the reason they are all considered "numerically oriented"; then we describe each of the MPs in some detail.

## The Lexical and Ontological Treatment of Scalar Attributes in OntoSem

Among the properties defined in the PROPERTY branch of the OntoSem ontology are SCALAR-ATTRIBUTES, a random sample of which (selected from dozens) includes COMPLEXITY, COST, INTENSITY, USEFULNESS, DURATION, 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 an abstract {0,1} scale. For example, if a car costs $13,500, this fact will be represented in the TMR as follows:

```
COST
    DOMAIN          AUTOMOBILE-334
    RANGE           13500
    MEASURING-UNIT  DOLLAR
```

By contrast, if a car is said to be "expensive", that fact is represented as follows:

```
COST
    DOMAIN          AUTOMOBILE
    RANGE           .8
```

*Expensive* is realized as .8 in the TMR because the lexicon entry for *expensive* specifies that *expensive* refers to .8 on the scale of COST. Of course, one could quibble about whether *expensive* should be .7, .8, (< > .7 .9), etc.; but lingering over such unresolvable questions does not support practical solutions. Therefore, during lexicon acquisition we attempt, in a naive way, to be consistent in our interpretation of points on the scale (just as *expensive* is .8 on the scale of COST, *tall* is .8 on the scale of HEIGHT, *heavy* is .8 on the scale of WEIGHT, etc.). Moreover, we assign numerical values with a view toward potential eventualities: e.g., since one can say *very expensive* and *extremely expensive* – which will be higher on the scale of COST than just *expensive* – we allow a buffer for modification in the numerical expression of the range of *expensive*.

Of course these scalar values between 0 and 1 have concrete meaning only if one knows what the general range of values for something is. That is, an *expensive car* implies a different amount of money than an *expensive jump rope* or an *expensive satellite.* We can reason about the actual values involved in such phrases based on information recorded in the OntoSem ontology. For these examples, we (or the OntoSem semantic analyzer) would look up the ranges of COST defined in the ontological concepts to which *car, jump rope* and *satellite* are mapped (these words happen to have univocal ontological mappings to the concepts AUTOMOBILE, JUMP-ROPE and SATELLITE, respectively), then calculate the range that represents 80-100% of the expensive extreme. For example, the typical, ontologically defined COST of an AUTOMOBILE is 10-80K,[1] so an *expensive car* is:

```
< > ((.8 * (size of range)) + (low value of range))   high value
< > ((.8 * (80-10)) + 10)   80
< > 66  80
```

Such calculations can also be extended to specific subtypes of objects (e.g., *expensive Cadillac* vs. *expensive Kia*), assuming that the OntoSem knowledge sources have the typical costs for each of these recorded.

This background on the general treatment of scalar attributes in OntoSem is relevant for all of the MPs described below.

## Delimit Scale

Delimit-scale is the MP that calculates the modified value of a SCALAR-ATTRIBUTE that is expressed as a point on the abstract {0,1} scale. It is placed in lexical entries for words like *very, extremely, quite, moderately, somewhat,* etc. For example, one lexical sense of *very* is shown below, in presentation format.

```
very-1
  cat adv
  def  "toward the more extreme end of the given scale"
  ex "very big, very late, very small"

syn-struc
     mods  $var0 (cat adv)
     root   $var1 (cat (or adj adv))

meaning-procedure
     delimit-scale (value ^$var1) extreme .1
```

The syn-struc (syntactic structure) says that *very* ($var0) is an adverb that modifies an adjective or an adverb ($var1). Unlike typical lexicon entries, this one has no static *sem-struc* (semantic structure) zone, since the meaning of *very* relies on its composition with what it modifies. Instead, it has a *meaning-procedure* zone that calls the delimit-scale MP with three arguments:

1. the value of the meaning of $var1 (indicated by ^$var1), which is a value between 0 and 1;

[1] We realize that there are cars that cost more than $80K. Inputs like *He bought a car for $350,000,* when found in a trusted source, might suggest the need for modification to the high end of the scale of CAR.COST. However, the analysis of *extremely* will be done on the basis of knowledge already available in the ontology. This situation clearly parallels human experience of learning new facts and concepts.

2. whether the scalar value is shifted toward the extreme or the mean of the given scale;
3. the amount by which the value is augmented.

So, *very small* would be calculated by taking the value of small (SIZE .2) and shifting it to the extreme by .1, returning a value of (SIZE .1). Analogously, *moderately small* would be calculated by taking the value of small (SIZE .2) and shifting it toward the mean by .1, returning a value of (SIZE .3). (The MP for *moderately* has the 2nd argument as 'mean' rather than 'extreme'.)

An interesting situation occurs if one modifies a scalar such that its value is off the scale. For example, *extremely* is defined as shifting the scalar value by .2 toward the extreme, so an *extremely extremely expensive car* will be calculated as follows:

extremely + extremely + expensive
.2 + .2 + .8 = 1.2

The value 1.2 lies outside of the scale defined as 0-1; however, this is exactly what we want as a semantic interpretation of *extremely extremely*. To understand why this is so, a few more words about the ontological encoding of property values are necessary.

Property values in the OntoSem ontology can be defined using a number of different *facets*, including *sem* (which represents typical selectional restrictions), *default* (which represents a more restricted, highly typical subset of *sem*), and *relaxable-to* (which represents an extended interpretation of *sem*). E.g., a DOG most typically eats DOG-FOOD (*default* facet), but is perfectly capable of eating any INGESTIBLE (*sem* facet), and can even eat NEWSPAPER, GRASS, CARPET... (*relaxable-to* facet). When fillers for properties are themselves concepts, we try to explicitly cover all eventualities as specifically as possible, as with the example of what dogs eat. However, when the property is a scalar and its values are, accordingly, a range on a scale, really only the *default* and *sem* facets need to be explicitly encoded because the *relaxable-to* values can be inferred by extending the values for the *sem* facet in either direction. This is exactly what is happening in the case of *extremely extremely expensive*: a perfectly valid calculation of 1.2 is being returned, which means that the value is not among our typical expectations (as reflected in the ontology[2]). Instead, it goes outside of our typical expectations, into what is conceptually *relaxable-to* values. If we assume, as before, that the cost range of a car is typically 10-80K, then an *extremely extremely expensive car* will cost 94K. If one does not agree that an extremely extremely expensive car is one that costs 94K, then one could change the numerical value for *extremely* in the lexicon (perhaps it should shift the base value by .25, not .2), or change the typical cost of cars in the ontology (perhaps the typical upper limit is 100K, not 80K), etc.

Before turning to the next MP, let us reiterate the reason for specifying these values in the first place: if one text says that the president of France bought a *very expensive car* using government funds, and another text says that the president of Russia bought an *extremely expensive car* using government funds, and a user asked our Q&A system "Who bought a more expensive car, the President of France or the president of Russia", the answer returned – based on the comparison of scalar values for COST – will be "the president of Russia" (who, by the way, our fact repository will understand to be Vladimir Putin).

## Decrease-/Increase-value

Decrease-value and increase-value are a pair of meaning procedures that, like *delimit-scale*, are used to calculate the modified value of a SCALAR-ATTRIBUTE. However, in contrast to delimit-scale, these MPs work on real (not abstract) quantities. For example, *nearly 10 years, almost a week* and *a little shy of 100 dollars* are more specifically resolved as *9.5 years, 6 days,* and *95 dollars*, respectively – all understood by the system as calculated values and, therefore, defeasible. Consider, for example, the lexical entry for the sense of *nearly* that modifies a count noun.

```
nearly-2
    cat adv
    synonyms "almost" "just-under"
    def "nearly + number + noun"
    ex "it's been nearly 10 years since I moved here"

 syn-struc
    root     $var0  (cat adv)
    num      root $var1 (cat num)
    n        root $var2  (cat n)

 meaning-procedure
    decrease-value (cardinality (value ^$var1)) .05
```

The *decrease-value* MP says that one must take the given value for cardinality, multiply it by .05 and subtract that product from the original cardinality. Thus, given the input *nearly 10 years*, the decrease-meaning MP will calculate (.05 * 10), yielding .5, then decrease 10 by .5, yielding 9.5. This will make the extended TMR for *nearly 10 years* look as follows:

```
DURATION
    DOMAIN     YEAR
    RANGE      9.5
```

The choice of .05 as opposed to, say, .06, or the range between .01 and .05, is, again, subject to amendment based on application- or domain-specific evidence. An example of when OntoSem would need such values for reasoning is as follows. Say a user asked a Q&A system to list all US senators who served for between 8 and 10 years; and say there is a text in the corpus that says that Senator X served for nearly 10 years; by resolving *nearly 10* to *9.5* the system will know to include Senator X in the list it returns. So, to reiterate, the reason for resolving scalar quantities at the level of extended TMRs is so that these numerical values can act as input to further reasoning programs.

## Specify-approximation

For lexical items that indicate approximations, like *about*, an extended-TMR-level representation requires creating a

[2] Expectations vary across different people and software agents with different ontologies. For the purposes of the current applications of OntoSem, we assume a simplified model with a common ontology between the speaker and the hearer (or the text author and the analyzer).

range with this number in the middle. The extent to which the range should be expanded in either direction is not possible to definitively specify, but we have found that for many cases 7% works pretty well. For example:

- About 5 gallons (5 * .07 = .35) is between 4.65 and 5.35 gallons
- About 150 lbs. (150 * .07 = 10.5) is between 139.5 and 160.5 lbs.
- About 8 hours (480 min. * .07 = 33.6) is between 7h.43.2m. and 8h.16.8m.

Thus, our current basic rule for calculating approximations is the 7% rule, with rounding to whole numbers when necessary (e.g., 8.7 people makes no sense). However, this rule is too coarse-grained on at least two grounds. First, the actual number from which the approximation derives is important in terms of what the approximation actually means. In most cases, one adds an indication of approximation to a round number like 10, 25, 100 or 5,000,000. It's odd to say *about 97 people* or *around 8.24 pounds.* If, however, someone did use such a turn of phrase, the interpretation of the approximation would be something different than what would be returned by the 7% rule. We did not pursue such pragmatically odd cases because they are, in practical terms, not of high priority.

There are, however, quite a number of commonly encountered cases where the 7% rule fails and where special, semantically targeted programs are needed. They include:

- *Heights of people*: Using the 7% rule, *about 6 feet tall* would give a range of over 5 inches on either side, which is far too broad. Moreover, people usually judge each other's heights within about an inch or an inch and a half no matter what height they are, so we should not be calculating a percentage of the total height but, rather, fixing "1-1.5 inches either way" as the range of approximation for human height (note that people judge each other's height with greater accuracy than, say, their weight).
- *Ages.* Interpreting the approximation of a person's age depends on how old the person is, with the 7% rule working poorly for children but better for adults. For example, the 7% rule would make a baby who is about 5 days old be 5 days +/- 8.4 hours, which is probably not what is intended when someone says *about 5 days old* (what is intended is more likely 4-6 days old). Likewise, a child who is about 5 years old would be 5 years +/- 3.5 months – again, quite a bit more fine-grained than would be intended. As a person gets older, however, the 7% rule works better: a person who is about 80 years old would be roughly 75-85, and a person who is about 50 years old would be 46.5-53.5. In this case, as in the last one, it seems more direct to simply set the buffer for the approximation of given age ranges rather than try to force the 7% rule – which is what we actually did.
- *Clock time.* The 7% solution is reasonable for "round" clock times, but much less so for more precise clock times. Rather than employ it, we are using a different approach to calculating approximate clock times:
  a) around [hour or ½ hour or 'noon', 'midnight'…] = +/- 10 min.
  b) around [1/4 hour] = +/- 8 min.
  c) around [divisible by 10] = 4 min.
  d) around [other] = 2 min.
- Temperatures are not handled well by the 7% rule because, no matter what the temperature is, the use of approximation tends to imply 5 or 10 degrees in either direction.

In short, there are many details hidden in the use of approximation that we have not yet pursued and will not pursue until we find a practical need for them: i.e., until our ability to automatically reason is hampered by not modifying the 7% rule. What we do not want to do in a practical system is show off our acuity as lexical semanticists, which could lead down an endless path of potentially unneeded research about approximation.

Our implementation of the approximation MP currently covers the 7% rule and all of the exceptions listed above, which are diagnosed using OntoSem semantic analysis.

## Final Thoughts

As mentioned earlier, the three meaning procedures described here represent only a sampling of MPs currently implemented in the OntoSem environment. Some MPs, like the ones discussed here, are triggered explicitly by calls from lexical items encountered in the text. Other examples of MP-triggering lexical items include (but are not limited to): a) **pronouns**; b) the article ***the*** (since the use of the definite article may or may not signal the need to corefer with an antecedent, and if it does, that antecedent must be determined); c) the determiners ***this/that/these/those*** (similar in resolution needs to *the*, but more complex since they might be part of a NP or might be independent NPs); d) **spatial and temporal indices**, like *today, there*, etc., and all complex NPs that include them (*two weeks from today*).

Within the OntoSem environment, lexical specification of MPs is only one way in which they are triggered. For example, there are MPs that attempt to reason using material that is not overt in the text but is available in the ontological descriptions of concepts instantiated in the TMRs. This sort of reasoning is done on an as-needed basis, e.g., when semantic ambiguity remains after the basic stage of analysis. No matter what the triggering condition, MPs seek to provide the semantic information necessary for high-end reasoning applications.

## References


Sergei Nirenburg, Marjorie McShane and Stephen Beale. 2003. Operative strategies in Ontological Semantics. *Proceedings of HLT-NAACL-03 Workshop on Text Meaning*, Edmonton, Alberta, Canada, June 2003.

Sergei Nirenburg and Victor Raskin. Forthcoming. *Ontological Semantics*, the MIT Press, Cambridge, Mass.