Achieving Adequacy of Description of Multiword Entities in Semantically-Oriented Computational Lexicons

Marjorie McShane, Sergei Nirenburg, Stephen Beale

Working Paper #01-08

2008

Institute for Language and Information Technologies

University of Maryland Baltimore County

Abstract: This article discusses three aspects of recording multiword expressions (MWEs) in semantically oriented lexicons for NLP: achieving syntactic adequacy, achieving semantic adequacy, and computing the semantic contribution of non-compositional elements. The purpose of the analysis is two-fold: first, to provide a descriptive, example-based account of how complex aspects of MWEs can be treated in computational lexicons; second, to bring attention to some aspects of MWEs that are not currently being treated by most systems but must be treated if we are to achieve truly sophisticated natural language processing.

Key words: multiword expressions, computational lexicons, computational semantics, idioms

1. Introduction

This article discusses three aspects of recording multiword expressions (MWEs) in semantically oriented lexicons for NLP: achieving syntactic adequacy, achieving semantic adequacy, and computing the semantic contribution of non-compositional elements. Whereas theoretical work has treated the first two of these, practical work has focused primarily only the first; and the third, to our knowledge, has been treated rather minimally from any perspective. Although the examples to be discussed are worked using a specific formalism from a particular environment, the analysis is system- and environment-neutral: any environment that includes syntactic and semantic analysis can benefit from the types of knowledge structures and processing rules described here. The background for the analysis reaches rather broadly and will be covered by a heterogeneous series of subsections in this introductory section.

1.1 What are MWEs?

Defining MWEs is tricky. In broad strokes, they are linguistic constructs composed of more than one word that are considered – for any number of reasons – to be better treated together than analyzed compositionally. This definition certainly includes expressions that, synchronically speaking, could as well have been single words, like *notary public* and *garden hose*. It also includes expressions that are unquestionably idiomatic, like *eat one's words* and *take a hike.* But there is a large conceptual territory around and between these islands of "definite MWEs" that can be occupied by various types of other expressions, such as phrasal verbs (*back up*, *back down*), collocations that contain (quasi-)light verbs (*take someone's temperature, get a physical*) and noun-noun compounds. As concerns the latter, in cases when the relationship between entities is fixed, it can be preferable to treat the compound as a MWE, recording the "frozen" semantics of the relationship explicitly: e.g., *dorm room* always means a room serving as living quarters in a dormitory. By contrast, since *IBM lecture* can be a lecture about IBM, sponsored by IBM, given by an employee of IMB, and so on, it would not be advisable to encode any of the meanings statically since an encoded meaning would, in most systems, be automatically preferred over competing compositional meanings. Our practical work has convinced us that there is no hard line between what should and should not be recorded as an MWE; instead, individual judgments about ease and accuracy of processing play a large role in making this determination.

1.2 Related Work

Recent work on MWEs has produced substantial results, particularly in the realm of automatically detecting new MWEs in corpora and improving parsing through MWE-supported lexicons. For an overview of methods applied to these tasks see Schone and Jurafsky 2001. The automatic detection of MWEs in corpora has been applied to various subtypes of MWEs, including verb-particle constructions (Baldwin and Villavicencio 2002; Kim and Baldwin 2006; among many others), collocations (Smadja 1993), and even any type of MWE in any language (Dias 2003). Preliminary thoughts about how to present both syntactic and semantic information about MWEs in multilingual lexicons are presented in Calzolari et al. (2002), though the discussion is heavily weighted toward classification and problems of MWEs rather than specific recommendations. For a broader overview of recent directions of work, see the contributions to the ACL 2004 workshop "Multi-word Expression: Integrating Processing" and the EACL 2006 workshop "Multi-word Expressions in a Multilingual Context"; see also papers produced by Stanford's Multi-Word Expression Project (http://mwe.stanford.edu/).

As regards idioms, which are a subtype of MWEs, a cornerstone of theoretical, descriptive, computational and psycholinguistic work has been the attempt to understand to what extent idioms are fixed and to what extent they are flexible. For discussion, see, e.g., Cacciari and Tabossi (1993), whose component articles provide particularly extensive and well selected reviews of the literature. The competing classifications can derive from theoretical considerations, like psycholinguistic evidence, or practical considerations, like whether an NLP system attempts to analyze only those idioms that are recorded, or whether it attempts to analyze new coinages as well.

The scope of the current analysis is all kinds of MWEs, idiomatic and not, that are recorded in the lexicon, as well as certain types of free modifications of them. Like Stock et al. (in Cacciari and Tabossi 1993, p. 238), we integrate MWEs into the lexicon as "more information about particular words" rather treat them using special lists and idiosyncratic procedures.

Let us briefly summarize the differences between the main currents of past work and the current contribution:

(a) Automatic MWE detection systems typically stop at detecting the entity, not semantically analyzing it or deducing its lexical or syntactic variations. In some cases (e.g., Sharoff 2004, Venkatsubramanyan and Perez-Carballo 2004), developers say that the goal is to pass off candidate MWEs to lexicographers or ontologists for manual incorporation into NLP resources. In the work described here, we carry out that next step, showing what lexicographers and programmers must do, in tandem, to fully treat MWEs both syntactically and semantically.

(b) Automatic MWE detection systems typically extract the types of entities that are most readily detected using the given methodology, not necessarily the full complement of MWEs. By contrast, we consider the needs of systems first then respond to them by developing representational means and static knowledge sources.

(c) Whereas automatic MWE detection systems can be readily evaluated using measures like recall and precision, the contribution of manually and semi-automatically compiled knowledge bases is much harder to evaluate. Clearly, having a large lexicon is better than having a small one, so with every new entry – be it for an MWE or a simple word – the lexicon improves. But only in an application can the lexicon be judged, and even then blame assignment is often not trivial. Therefore, let us say from the start that the analysis we offer is not evaluated, though we are incorporating MWEs into our lexicon and they are being leveraged as we build and test applications.

1.3 The Role of Manual Acquisition in Resource Development

The creation of computational lexical resources has benefited from the work of lexicographers who have, over the centuries, been compiling mono- and multi-lingual, human-oriented lexicons. There are several modes of exploiting human-oriented lexical resources for NLP:

1. If the resource is digital, use it directly, as has been done, for example, with WordNet (http://wordnet.princeton.edu/). Features to enhance the resource's utility for NLP can then be added as resources permit. See, e.g., Harabagiu et al. (1999) for NLP-oriented extensions to WordNet, and Fellbaum (1998) for ideas about how to improve the coverage of idioms in WordNet.
2. If the resource is not digital, digitalize it and proceed as in (1). This approach was actively pursued in the 1990s in a push to exploit so-called machine-readable dictionaries. However, the results were disappointing and that program of work has largely been abandoned. See Ide and Véronis (1993) for a critical analysis, and Inkpen and Hirst (1999) for later efforts to exploit machine-readable dictionaries.
3. Have people exploit human-oriented resources to speed up the manual or semi-automatic creation of resources for NLP. This is the approach taken in the OntoSem environment to be described here.

The main objection to the third strategy involves resources: this mode of acquisition requires the most time and money. One way of justifying the outlays is to have a resource to do double duty. Is this really possible? We will suggest it is, providing as evidence a human-oriented lexicon that has many of the features of lexicons oriented toward NLP.

The dictionary in question is the *Random House Russian-English Dictionary of Idioms* (Lubensky 1995), which is a large (over 1000-page) learner's dictionary that includes, for each sense of each idiom, grammatical usage information, stylistic usage information, all lexical and syntactic variations of each idiom, a definition, sample patterns of usage, sample English translations, and literary examples along with their contextually sensitive English translations. A example of one sense is shown below, first in Cyrillic then in transliteration.[1]

> Б-58 НАША <ВАША, ТВОЯ> БЕРЕТ/ВЗЯЛА *coll* [VP_{subj}; more often pfv; if impfv, pres only; fixed WO] we (you etc) are (or are about to be) victorious: наша взяла ≃ **we've won; our side (has) won; we're the winner(s) <victor(s)>;** || наша берет/возьмет ≃ **we're winning <going to win>; our side is winning <going to win>; we're going to be the winner(s).**
> …Я чувствовал, что за эту манеру стоять, повернувшись спиной возле самого носа, ломаться и показывать «мы-де победители – наша взяла» следовало бы их всех бросить в воду… (Герцен 2). …I felt that in return for their manner of standing and turning their backs in our very faces, giving themselves airs and showing off, "We are the victors – our side won, " they ought all to have been thrown into the water… (2a) ♦ «„Дураки, – говорю я им, – глупые несмышленыши. Эту власть Гитлер не смог опрокинуть со своими танками, а вы что сможете со своей болтовней?.. " – „Ничего, – говорит один из них, это так кажется, что они сильные, наша возьмет"» (Искандер 4). "'You fools,' I said to them, 'you're a couple of babes in the woods. Hitler couldn't topple this regime with all his tanks, and what can you do with your blather?...' 'Nuts,' says one of them. 'They only look strong, we're going to win'" (4a).

> B-58 Nasha <vasha, tvoja> beret/vzjala *coll* [VP_{subj}; more often pfv; if impfv, pres only; fixed WO] we (you etc) are (or are about to be) victorious: nasha vzjala ≃ **we've won; our side (has) won; we're the winner(s) <victor(s)>;** || nasha beret/voz'met ≃ **we're winning <going to win>; our side is winning <going to win>; we're going to be the winner(s).**
> …Ja chustvoval, chto za ètu maneru stojat', povernuvshis' spinoj vozle samogo nosa, lomat'sja I pokayvat' «my-de pobediteli – nasha vzjala» sledovalo by ix vsex brosit' v vodu… (Hertzen 2). …I felt that in return for their manner of standing and turning their backs in our very faces, giving themselves airs and showing off, "We are the victors – our side won, " they ought all to have been thrown into the water… (2a). ♦ «„Duraki, -- govorju ja im, -- glupye nesmyshlenyshi. Ètu vlast' Gitler ne smog oprokinut' so svoimi tankami, a vy chto smozhete so svoej boltovnej?.." -- „Nichego, -- govorit odin iz nix, èto tak kazhetsja, chto oni sil'nye, nasha voz'met''» (Iskander 4). "'You fools,' I said to them, 'you're a couple of babes in the woods. Hitler couldn't topple this regime with all his tanks, and what can you do with your blather?...' 'Nuts,' says one of them. 'They only look strong, we're going to win'" (4a).

[1] Indications of word stress have been excluded.

This is the 58th idiom whose main head word starts with the letter Б (B). The variations of the first word (which means *our* <$\textit{your}_{\textit{PLURAL/POLITE,}}$ $\textit{your}_{\textit{SINGULAR}}$>) are shown in pointed brackets. The tense and aspect variations on the verb ($\textit{takes}_{\textit{IMPERFECTIVE.PRESENT}}/\textit{took}_{\textit{PERFECTIVE.PAST}}$) are shown using slash notation. Stylistically, the idiom is colloquial (*coll*). Syntactically, it contains both the subject and the predicate ($\mathrm{VP}_{\mathrm{SUBJ}}$). The perfective (*pfv*) aspect is used more often than the imperfective (*impfv*) aspect. If the imperfective is used, the verb must be in the present tense. The word order of the idiom is fixed. All of this information is presented in a strict formalism that could be automatically converted into processing rules for parsing or language generation. The remaining sections of the entry could also be exploited for NLP, but more work – either involving machine learning, manual intervention or both – would be required.

The non-grammatical section begins with a definition that might include usage notes in parentheses. Next come translations that in some cases – as in our example – are associated with patterns of usage. After the translations are examples, most of which are drawn from literary works, along with their English translations. One can easily imagine the types of machine learning that could be applied to such a corpus of parallel Russian-English examples that contain a given idiom.

The relevance of this dictionary for NLP is as follows: Non-native speakers of a language and computer programs need to know the same things about idioms. If this information is encoded in a formal way, it can directly be exploited by both "consumers". Of course, formal descriptions that can do double duty are more easily achieved at the level of lexico-syntax than the level of semantics since people would arguably not want to read formal representations of the meaning of text, and computer systems struggle with the ambiguity of natural language until that ambiguity has, in one way or another, been resolved (the latter was, of course, the main pitfall of using machine-readable dictionaries for NLP).

A good way to illustrate the difference between how people interpret the semantics of lexical entries and how computers interpret it is to look at actual entries of human-oriented lexicons. Consider, for example, an excerpt from the entry for the phrasal "go down" in dictionary.com (all senses that are presented are presented in full; senses 5-7 have been omitted):[2]

1.
    a. To drop below the horizon; set: *The sun went down.*
    b. To fall to the ground: *The helicopter went down in a ball of fire.*
    c. To sink: *The torpedoed battleship went down.*
    d. To experience defeat or ruin.
2. To admit of easy swallowing: *a cough syrup that goes down readily.*
3. To decrease in cost or value.
4. Chiefly British To leave a university.

[2] http://dictionary.reference.com/search?q=go

Now consider how much real-world knowledge a person brings to the interpretation of these sensesd:

- Sense 1a can be used exclusively for the sun, not a ship going into the horizon, or a building as viewed from a train moving away from it.
- Sense 1b can be used of flying vehicles but, typically, not of leaves or other falling objects.
- Sense 1c should be constrained to vessels, because if a fish goes down, the implications are generally quite different: it swam down intentionally.
- Sense 3 is actually not quite precise: it is intended for materials (gold) and possibly other mass nouns (trust), whereas it does not work for class nouns. Indeed, if a car decreases in cost or value, it cannot be said that the car went down – it is the *cost* of the car that went down. So this sense would more precisely be rendered "(of the cost or value of something) to decrease" (Nirenburg et al. 2005).
- Sense 4 is not sufficiently specified for unambiguous interpretation, since "leave" can mean simply go away from, as in "The truck left [but not *went down from*] the university after dropping off the textbooks".
- In all senses, the 'go' in the head phrase 'go down' can undergo normal inflection.

Since people – at least those with a reasonable command of English, for whom dictionary.com is likely intended – can easily make the necessary inferences, there is no need to clutter up the dictionary with further explanations: scanability and conciseness are at a premium. However, real interpretation by machine requires much more highly specified lexical description, as we will show in the remainder of the paper.

To conclude this discussion, a dictionary containing the information of the Russian-English dictionary described here could be used directly for NLP for lexical and syntactic processing. However, when it comes to semantics, no matter what human-oriented lexicon one uses, manual intervention is needed to convert the information into a form that can be unambiguously interpreted by computer systems.

1.4 An Environment for Pursuing the Comprehensive Processing of MWEs

Within the field of NLP, systems that are devoted specifically to deriving the deepest levels of meaning from texts are quite rare. Typical application systems use relatively knowledge-lean approaches that rely on sophisticated clustering algorithms that circumvent the need to semantically analyze and disambiguate text. As concerns lexical knowledge, most systems use a word net (e.g. WordNet and its non-English spinoffs) or some other large compendium of lexical items that are described by few properties. Such systems are quite sufficient for some applications, as even the most cursory review of the literature attests. However, they will arguably never support truly high-level, reasoning-intensive ones, like creating an intelligent agent with the smarts of Star Trek's *Data* or *Hal* from the movie 2001: A Space Odyssey. Developers of knowledge-lean systems do not have such aims in mind: typically, they are interested in either creating near-term applications with available knowledge resources and technologies, or in developing NLP methodologies themselves, with the applications being of secondary importance.

Our group, by contrast, is interested in automatically achieving deeper analysis of meaning, despite the challenges of coverage and the virtual impossibility of carrying out the types of large-corpus evaluations that have become the standard. Our framework is the theory of Ontological Semantics (Nirenburg and Raskin 2004), as implemented in the OntoSem system. OntoSem takes as input unrestricted text and carries out its tokenization, morphological analysis, syntactic analysis, and semantic analysis to yield text-meaning representations (TMRs), which are disambiguated, formal representations of meaning encoded in a language-independent metalanguage. Text analysis relies on:

- The OntoSem language-independent **ontology**, which uses its own metalanguage and currently contains around 9000 concepts, each described by an average of 16 properties ("features") selected from the hundreds of properties defined in the ontology. The number of concepts is intentionally restricted so that mappings from lexicons are many to one; further meaning specification of words and phrases is done in the lexicon.
- An OntoSem **lexicon** for each language processed, whose entries contain, among other information, syntactic and semantic zones that are linked through special variables, as well as procedural-semantic attachments that 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 and time. An extensive tutorial for lexicon developers can be accessed at http://ilit.umbc.edu.
- A **fact repository**, which contains real-world facts represented as numbered "remembered instances" of ontological concepts: e.g., SPEECH-ACT-3186 is the 3186th instantiation of the concept SPEECH-ACT in the world model constructed during text processing as the embodiment of text meaning.
- The OntoSem text **analyzers**, which cover tokenization, morphological, syntactic and semantic analysis, and the creation of text-meaning representations (TMRs).
- The TMR language, which is the **metalanguage** for representing text meaning throughout the OntoSem environment.

Below is an example of a TMR, in a simplified presentation format, for the sentence **Colin Powell was appointed Secretary of State**. We use a toy example as it will suffice for illustration and is easier to follow than the multi-page TMRs for longer sentences.

```
SOCIAL-EVENT-1
    textpointer     appoint
    EFFECT          HAS-SOCIAL-ROLE-1
    time            < find-anchor-time
HAS-SOCIAL-ROLE-1
```

```
    DOMAIN                  HUMAN-1
    RANGE                   SECRETARY-OF-STATE-1
    CAUSED-BY               SOCIAL-EVENT-1
HUMAN-1
    textpointer             Colin_Powell
    HAS-PERSONAL-NAME       Colin
    HAS-SURNAME             Powell
    DOMAIN-OF               HAS-SOCIAL-ROLE-1
    FR-REFERENCE            HUMAN-FR24
SECRETARY-OF-STATE-1
    textpointer             Secretary_of_State
    RANGE-OF                HAS-SOCIAL-ROLE-1
```

Instances of ontological concepts are numbered. If this had been a text in the middle of a large corpus, the numbers of the concept instances would have been much higher. The time slot filler "< find-anchor-time" is a call to a procedural semantic routine that seeks the time the report was made – as might be found in a dateline, in metadata, or in the previous context – and asserts that this event happened before that. Disambiguation has been carried out: *appointed* is interpreted as a SOCIAL-EVENT whose EFFECT is that the given person HAS-SOCIAL-ROLE as indicated. The analysis process has excluded the other meanings of *appoint*: 'fixed by mutual agreement' and 'equip'. This interpretation of the word *appoint*, like the interpretations of all lexical items, was explicitly recorded in the lexicon. Apart from wordsense disambiguation, the OntoSem analyzer has resolved reference against its repository of stored real-world knowledge, called the Fact Repository. The Fact Repository already had information about a person with the last name Powell, who is stored under the key HUMAN-FR24. Since the information about Powell in this text did not contradict that stored knowledge, coreference was assumed and this new information was added to that previously acquired information.[3] In fact, populating the Fact Repository with information from TMRs, then using the Fact Repository as a search space for applications, is the main approach to system development in OntoSem.

OntoSem lexicon entries are written in Lisp-compatible format following a significantly extended version of Lexical Functional Grammar. A sample OntoSem lexicon entry for a transitive verb (slightly simplified for presentation) is as follows:

```
(watch-v1
    (def "to engage in a voluntary visual event")
    (ex "He was watching the demolition team.")
    (syn-struc
        ((subject ((root $var1) (cat n)))
         (root $var0) (cat v)
         (directobject ((root $var2) (cat n) (opt +)))))
    (sem-struc
        (VOLUNTARY-VISUAL-EVENT
```

[3] The process of establishing coreference is actually more involved, but the details go beyond the scope of this article.

```
        (AGENT (value ^$var1))
        (THEME (value ^$var2)))))
```

The syntactic structure (syn-struc) says that this is an optionally transitive sense of *watch*, and the semantic structure (sem-struc) says that it refers to a VOLUNTARY-VISUAL-EVENT, which is a concept in our ontology. The variables are used for linking: the syntactic subject is linked to the meaning of the AGENT of the VOLUNTARY-VISUAL-EVENT (^ is read 'the meaning of'), and the syntactic direct object is linked to its THEME.

When writing syntactic structures for non-MWE entries, like garden variety transitive, bitransitive or intransitive verbs, we refer to syntactic functions like subject, direct object, indirect object, xcomp (a verbal complement headed by an infinitive), comp (a clausal complement headed by an optional 'that'), prep-part (a verbal particle that, in form, looks like a preposition), etc., as in the example above. The constituents can be written in any order, since their actual ordering is understood using a few basic rules in our parser supplemented by an inventory of transformations similar to the kind found in XTAG (www.cis.upenn.edu/~xtag).[4]

By contrast, when writing syntactic structures for MWEs, we often refer to immediate constituents, like NP, N, Adj, Prep, Conj, etc. There are two reasons for this: first, the syntactic structure of many MWEs is idiosyncratic. Our formalism permits us to specify any sequence of elements, including punctuation, without the requirement of creating from them canonical "well formed" syntactic structures. The second reason for including immediate constituents in a syntactic structure involves blocking transformations in cases when they must not be applied. The ordering of constituents in syn-strucs containing one or more immediate constituents is understood as being fixed, and the structure is not subject to transformations.[5] Therefore, if we want to block transformations, even on canonical syntactic structures, we can include at least one immediate constituent label and the transformations will automatically be blocked.

2. Fixed Versus Composed MWEs

For practical purposes, one can delineate two broad classes of MWEs: those that can be treated as fixed entities that happen to have a white space between them, and those that require specialized MWE treatment.

MWEs that fall into the first class include entities like *vice president* and *stock market*. The components of such MWEs must occur in the listed order, do not permit modifiers or other elements between them, and none of the components – except, potentially the last one – is subject to inflection. In short, a parser can interpret these as long strings with a space inserted at one or more points.

[4] Using syntactic function labels also supports porting of "typical" lexicon entries among languages, since properties of subject, direct object, etc., will be dealt with globally by the syntactic analyzer for each language. MWEs, by contrast, tend to be idiosyncratic for each language, so the decreased portability due to the use of immediate constituent labels represents a relatively minor loss.

[5] For English, fixing the ordering of components in syntactically idiosyncratic MWEs tends to work fine since syntactic transformations tend not to apply to such entities. If transformations are expected, more senses specifying permitted transformations must be supplied.

In the OntoSem lexicon, we record such entities as multi-part head words with an underscore indicating each white space. This approach provides simplicity and nearly perfect coverage -- only "nearly" perfect because in rare cases an expletive, speaker correction, interruption, etc., might occur between the elements, something that can also happen, by the way, in the middle of regular words: "deconstru [achoo!] ctionism." As with all so-called unexpected input, such deviations must be handled by recovery procedures which, in our system, amount to a sequence of attempts to loosen certain constraints, like the expectation that only a blank space can intervene between components of a multi-word head entry.

All other MWEs are recorded under a single headword, with the other elements being composed in the body of the entry. Composing MWEs in the body of lexicon entries provides for many expressive means and permits language-wide analysis processes to be applied to the resulting structures. For example:

- Non-head elements of MWEs can be lexically stipulated in the syntactic zone of the entry, along with any necessary grammatical features: *kick the bucket*, recorded under *kick,* requires its direct object to contain the definite article and the word *bucket* in the singular.
- Non-head elements of MWEs can be semantically constrained in the semantic zone of the entry: *go down*, recorded under *go*, means 'sink' only when the subject refers to a water vehicle.
- The syntactic structure of a MWE can be canonical (*kick the bucket* is a typical transitive construction) or it can be rather atypical (*X cannot help but Y)*. If it is atypical, the lexicon entry includes an indication of how the entity as a whole should be interpreted in the larger syntactic structure: e.g., *X cannot help but Y* should be analyzed as a clause. Asserting the top-level grammatical function of syntactically atypical MWEs saves us from spending undue time trying to impose a more canonical internal structure – especially since the internal structure is, for all intents and purposes, moot anyway.
- Whenever MWEs represent canonical syntactic structures, the parser's inventory of transformations are understood to apply in the normal way, unless explicitly blocked: *X attracted Y's attention to Z* can be passivized as *Y's attention was attracted to Z (by X)*. Methods for blocking transformations are discussed below.
- Components of MWEs listed in the body of lexicon entries are understood to permit their usual kinds of modification: *kick the bucket* can be modified to *kick the bloody bucket*. For a discussion of the interpretation of such modifiers, see Section 5.2.
- Components of MWEs listed in the body of lexicon entries are understood to undergo their usual kinds of morphological processes, unless explicitly blocked: *He would have kicked the bucket* is fine, but not *He kicked the buck-****ets*** since *bucket* has the feature 'number sing' in the syn-struc of the corresponding entry.

The work presented in this paper differs from other contributions in that it focuses not only on the semantics of MWEs (some aspects of which are treated, e.g., in

O'Hara and Wiebe 2003) but on particularly difficult aspects of the combined syntactic and semantic analysis of MWEs, including cases in which a non-compositional element carries certain semantic features that must be retained in the meaning representation of the entity.

3. The Syntax of MWEs

As just explained, MWEs in the OntoSem lexicon can contain any sequence of elements. Consider the lexical sense for the MWE *X {can} {not} {help} but Y,* which is recorded as a verbal sense of *help.* (We use { } to show words that can take various inflectional or other forms, as *not* being realized as *n't*.) Annotations for each line of the lexical entry are provided in gray italics.

```
(help-v4
   (def "the phrasal: X cannot help but Y" – X feels he must do Y because he cannot
force himself not to Y)
   (ex "The people in the room could not help but laugh. I cannot help but think to
myself that something smells fishy.")
   (syn-struc
      ((subject ((root $var1) (cat np)))              ;; the subject
       (v ((root $var2) (cat v) (root can)))          ;; the word 'can' with any inflection
       (verb-neg ((root $var3) (cat verb-neg)))       ;; verbal negation
       (root $var0) (cat v) (form infinitive)         ;; the bare infin. form of 'help'
       (conj ((root $var4) (cat conj) (root but)))    ;; the word 'but'
       (inf-cl ((root $var5) (cat inf-cl))))          ;; a bare-infinitive clause
   (output-syntax cl)                                 ;; this MWE functions as a clause
   (sem-struc
      (^$var5                               ;; The meaning of the bare-infinitive clause
          (agent (value ^$var1)))           ;; The AGENT is the meaning of the subject
      (^$var2 (null-sem +))              ;; The meaning of $var2 should not be computed
      (^$var3 (null-sem +))              ;; The meaning of $var3 should not be computed
      (^$var4 (null-sem +))))            ;; The meaning of $var4 should not be computed
   (meaning-procedure
      (fix-case-role (value ^$var1) (value ^$var5))))  ;; see footnote6
```

Although the syn-struc shows strong syntactic constraints, it nevertheless covers a wide variety of input sentences because:

i. The subject, like any noun phrase in lexicon entries, can be of any form or complexity: [*Even the people in the room who had come in late and really*

[6] Meaning procedures are procedural semantic routines that compute contextual meaning. This one is used when we know that the case-role indicated might not be appropriate for all events that might occur in the structure: e.g., whereas one can be an agent of LAUGH one cannot be an agent of DIE – one is an EXPERIENCER of DIE. The analyzer checks the case-role inventory of the given event and makes modifications, if necessary.

didn't know what was going on]$_{\text{SUBJ}}$ *could not help but laugh*. The analyzer uses a small inventory of rules, in combination with the lexicon entries of the words in the input, to build up the necessary structure (see Beale et al. 2003 for the details of syntactic/semantic analysis).

ii. The verb *can* can have various inflectional forms, with "inflection" understood broadly to include number, tense, mood and aspect: *They can not <could not, etc.> help but laugh.*

iii. The bare-infinitive clause can take any shape that is legal in English given the selecting verb in question (cf. (i)): *He could not help but laugh out loud <make a quick retort, fall in love with her, lavish his daughter with gifts>*.

iv. Various adverbs can be added: *They really could not help but laugh.*

This example is particularly good for showing the expressive power of MWEs on the syntactic side. On the semantic side, however, it is less typical, since the acquirer has decided to render the semantics of the whole MWE as just the semantics of the bare-infinitive clause, whose agent is the meaning of the subject. So, the sentence *He could not help but laugh* basically reduces to *He laughed*, which is represented in the text-meaning representation generated by the OntoSem analyzer as follows (this it the 3rd instance of LAUGH and the 34th instance of HUMAN in this run of the OntoSem analyzer):

```
LAUGH-3
   AGENT    HUMAN-34
```

One might ask, if OntoSem claims to produce fine-grained semantic analysis, why is the semantics of '{can} {not} {help} but' ignored? The reason is practical: how, exactly *would* or *could* one describe the semantic nuances contributed by this portion of the MWE? One might try something like: "Irrespective of whether or not X wanted to do Y, X did Y because, given some unspecified properties of Y, it would have been too difficult for X not to do Y."[7] However, even if this were deemed a reasonable analysis of the given nuances, this is still little more than an English paraphrase, which is still many steps away from being a formal representation that could support useful automated reasoning. The formal representation would be quite complex and it is not clear what goal it would serve. As mentioned earlier, we pursue practical applications rather than flexing our knowledge representation muscles, and our primary interest currently is populating a repository of disambiguated, real-world information to use as the search space in practical applications. Nuances as fine grained as this one go beyond both our manpower for knowledge acquisition and the needs of our applications. In fact, the needs of applications are the ultimate criterion for practical judg-

[7] Something simpler, like "X did Y although X did not want to" carries incorrect implications, and seems more suited to a context like "He laughed because it was his boss who was telling the joke and he didn't want to get fired."

ments about grain size of description, as argued in Nirenburg and Raskin 2004, chapter 9.[8]

Although we do not commonly reduce out aspects of the semantics of MWEs, the ability to do this is actually very useful, since the "padding" in some MWEs really contributes little to the sense of the whole. More examples of such reductions are the following, all of which resolve to the meaning of just the main proposition, X ('meaning of X' is written ^X):

1) the fact that X → ^X
The fact that you've got a cold is sad → *^[[you've got a cold] is sad]]*

2) it {turn} out that X → ^ X
It turned out that he was right all along → *^[he was right all along]*

3) it {be} just that X → ^X
It's just that I don't want to go to school → *^[I don't want to go to school]*

4) the month of X → ^X (same for 'year of', etc.)
He was born in the month of January → *^[He was born in January]*

To emphasize, we are not saying that these MWEs add no meaning to their propositions or that the meaning they add defies encoding in our lexicon. What we *are* saying is that the semantic contribution is so subtle, and the grain-size of analysis required so fine, that we are not, given our current focus of research, currently pursuing it. It is, however, important to record such MWEs in the lexicon so that they are syntactically parsed correctly, and so that the semantic analyzer is explicitly told (using the feature "(null-sem +)") which elements of MWEs do contribute compositionally to the meaning of the proposition and which ones do not.

4. The Semantics of MWEs

To further describe the OntoSem approach to lexically composed MWEs we will use selected examples from the more than 40 senses of the verb *go* in the OntoSem lexicon of English, most of which are MWEs. A pressing need, when one has many senses of a lexical item, is to ensure that the analyzer can select the correct one for the given context; otherwise, one ends up with rampant ambiguity, which is one of the most daunting problems in all of NLP.

As was shown above in the entry for *X cannot help but Y*, specific lexical items can be associated with components of the syn-struc: in that example, $var2 must have the root *can*, and $var4 must have the root *but.* Specifying component strings is, in fact, a typical way of creating MWEs in OntoSem.

[8] Of course, there are ways of preserving such nuances in applications like MT, as by adding a transfer-based component that explicitly included idiom-to-idiom correspondences.

Another way of facilitating wordsense disambiguation during text processing is to semantically constrain verbal arguments. For example, the phrasal verb *go down* can have different meanings depending on the meaning of the subject. So, if a WATER-VEHICLE goes down, it means that it is the THEME of a SINK event (go-v17 below), whereas if an AIR-VEHICLE goes down, it is the THEME of a FALL-AND-HIT event (sense go-v18 below).[9] Thus, the combination of syntactic structure (subject + *go* + *down*) and the semantic class of the subject (WATER-VEHICLE or AIR-VEHICLE) determines which meaning is intended.

```
(go-v17
    (def "phrasal: go down; of air vehicles only - to fall from the sky and hit the ground"))
    (syn-struc
        ((subject ((root $var1) (cat n)))
         (root $var0) (cat v)
         (prep-part ((root $var2) (cat prep) (root down)))))
    (sem-struc
        (FALL-AND-HIT
            (THEME (value ^$var1) (sem AIR-VEHICLE)))
        (^$var2 (null-sem +))))

(go-v18
    (def "phrasal: go down; of water vehicles only - to sink"))
    (syn-struc
        ((subject ((root $var1) (cat n)))
         (root $var0) (cat v)
         (prep-part ((root $var2) (cat prep) (root down)))))
    (sem-struc
        (SINK
            (THEME (value ^$var1) (sem WATER-VEHICLE)))
        (^$var2 (null-sem +))))
```

Another pair of lexical senses that are disambiguated using semantic constraints are go-v22 and go-v23: If the meaning of the subject of *{go} into* is of the ontological type BROADCAST or PRINTED-MEDIA (*The article went into the reasons for the conflict*), then the interpretation is that the given instance of BROADCAST or PRINTED-MATTER has the property ABOUT-AS-TOPIC filled by the meaning of the direct object. If, by contrast, the meaning of the subject is HUMAN or an ontological descendant of this concept (*The politician went into the reasons for the conflict*), then the interpretation is that the given HUMAN is the AGENT of a DESCRIBE event whose THEME is the direct object of the input.

```
(go-v22
```

[9] There are many other meanings of *go down* whose encoding include similarly narrow semantic constraints.

```
    (def "phrasal ‘go into’; the subject is a document or broadcast – to be about")
     (ex "That document goes into the reasons why this happened.")
    (syn-struc
        ((subject ((root $var1) (cat np)))
         (root $var0) (cat v)
         (prep-part ((root $var2) (cat prep) (root into)))
         (directobject ((root $var3) (cat np)))))
    (sem-struc
        (^$var1 (sem (or BROADCAST PRINTED-MEDIA))
            (ABOUT-AS-TOPIC (value ^$var3)))
        (^$var2 (null-sem +))))

(go-v23
    (def "phrasal: ‘go into’; subject is human – to describe")
     (ex "He went into the reasons why this happened."))
    (syn-struc
        ((subject ((root $var1) (cat n)))
         (root $var0) (cat v)
         (prep-part ((root $var2) (cat prep) (root into)))
         (directobject ((root $var3) (cat n)))))
    (sem-struc
        (DESCRIBE
            (AGENT (value ^$var1))
            (THEME (value ^$var3)))
        (^$var2 (null-sem +))))
```

An appropriate question at this point would be, “Where did the semantic constraint, HUMAN, on ^$var1 come from?” The answer: from the ontology. The sem-struc of go-v23 is headed by DESCRIBE, whose AGENT is *ontologically* constrained to HUMAN. Therefore, even though there is no explicit semantic constraint on the meaning of the AGENT in the *lexicon*, there is such a constraint in the ontology, which is always leveraged in combination with the lexicon during text processing. The main point to take away from this example is that lexical knowledge supplements ontological knowledge, and both are relied on equally during the analysis process. In other words, there is no rigid boundary between the sem-struc zone of the OntoSem lexicon and the OntoSem ontology. The choice about where to encode information is sometimes difficult but there are really no right and wrong answers and there are few consequences of making a choice that some might consider non-optimal: after all, if the needed information is provided *somewhere* in the environment, the analyzer can find it. Not promoting each acquisition choice to the level of theoretical import is crucial for supporting practical progress in a complex environment.

The same richness of semantic description is available for MWEs as for all other entities in the OntoSem lexicon. Apart from expressing meaning through a direct or modified ontological mapping, meaning can be expressed using extra-ontological descriptors, like values of aspect or mood, or by a call to a procedural semantic routine. We provide examples of each in turn.

Aspect. When *{go} out* is used of cigarettes, matches, fires, etc., it means to stop burning. This is analyzed as the concept BURN, whose THEME is the meaning of the subject of *{go} out,* and whose phase (one subtype of aspect) is "end". Although practically anything flammable can burn and then stop burning, not every flammable thing can "go out" in English – this phrasal is reserved for a nest of words ontologically described as FIRE, SMOKING-DEVICE and FIRE-STARTING-DEVICE. The corresponding semantic constraints are attached to ^$var1.

```
(go-v22 (cat v)
   (def "phrasal 'go out'; of fires, smoking devices or fire starting devices –
        to become extinguished")
    (ex "The cigarette went out")
   (syn-struc
      ((subject ((root $var1) (cat np)))
       (root $var0) (cat v)
       (prep-part ((root $var2) (cat prep) (root out)))))
   (sem-struc
      (BURN
         (THEME (value ^$var1) (sem FIRE SMOKING-DEVICE
                                       FIRE-STARTING-DEVICE)
         (phase end))
      (^$var2 (null-sem +))))
```

Modality. Modalities scope over the meaning of a proposition. OntoSem recognizes eleven types of modality – epistemic, belief, obligative, permissive, potential, evaluative, intentional, epiteuctic (indicating degree of success), effort, volitive, saliency. The values of all of these are indicated by decimal values on the abstract scale {0,1}. The MWEs in the examples below are semantically analyzed using *belief* and *obligative* modalities, respectively. The meaning of *it {is} expected that* (expect-v4) is, "There is a set of unnamed people who strongly believe that the given event will happen". This example also shows how pleonastic *it* is handled as a matter of course in our environment – it is simply attributed null semantics.

```
(expect-v4
   (def "phrasal: it *be* expected that...")
   (ex "It is expected that everything will turn out fine")
   (syn-struc
      ((subject ((root $var1) (cat n) (type pro) (root it)))
       (v ((root $var2) (root *be*) (cat v)))
       (v ((root $var0) (cat v) (form past-participle)))
       (comp ((root $var3) (cat v)))))
   (sem-struc
      (modality
         (type belief)
         (value 0.8)
         (scope (value ^$var3)))
```

```
        (^$var1 (null-sem +))
        (^$var2 (null-sem +)))))
```

The meaning of *Z {is} for X to Y* (for-prep10) is, “X must Y Z”, in which the obligative modality with a value of 1 scopes over the main proposition. In addition, the linking of syntactic and semantic variables is non-canonical and must be specified: the subject is typically the THEME of the main event and the object of the preposition is the AGENT.

```
(for-prep10
    (def "phrasal: Z is for X to Y; indicates who must be the agent of the event")
     (ex "This problem is for the chairman to resolve.")
    (syn-struc
        ((subject ((root $var1) (cat n)))
         (v ((root $var2) (root *be*) (cat v)))
         (pp ((root $var0) (cat prep)
              (obj ((root $var3) (cat np)))))
         (inf-cl ((root $var4) (cat inf-cl)))))
    (sem-struc
        (modality
            (type obligative)
            (value 1)
            (scope (value ^$var4)))
        (^$var4
            (agent (value ^$var3))
            (theme (value ^$var1)))
        (^$var2 (null-sem +)))))
```

Procedural semantics. Procedural semantics is a cornerstone of OntoSem text processing. It operationalizes the well-known need to interpret semantics *in context* (see McShane et al. 2004 and Nirenburg et al. 2003 for discussion). Calls to procedural semantic routines – what we call meaning procedures – can be used either in conjunction with static semantic representations or in lieu of them.

The MWE *{do} so* is a verbal pro-form, meaning that its actual meaning can only be restored by linking to a textual or extra-textual coreferent. Sense do-v5 covers the syntactic structure *X {do} so*.

```
(do-v5
    (def "phrasal: do so")
    (ex "He asked if I had finished mowing the lawn and I said I had done so")
    (syn-struc
        ((subject ((root $var1) (cat np)))
         (root $var0) (cat v)
         (adv ((root $var2) (cat adv) (root so)))))
    (sem-struc
        (EVENT
            (AGENT (value ^$var1))))
        (^$var2 (null-sem +)))
    (meaning-procedure
        (seek-specification (value $var0) reference-procedures)))
```

The coreferential event might occur earlier in the same sentence (separated by punctuation or not), or in earlier sentences. Just as with regular pronouns, one cannot predict beforehand exactly where it will occur, so rather generalized routines are needed to seek out the necessary coreferent. Those routines are encoded in the "seek-specification" meaning procedure. Its first argument is the meaning of the generalized EVENT that we posit in the sem-struc, which is locally correct but contextually insufficient. Its second argument indicates that our general inventory of coreference routines should be applied since this MWE does not provide any special heuristics for reference resolution.

5. Compositional Aspects of Non-Compositional Elements

In this section, we discuss how we circumvent two different problems that could arise from applying null semantics ("null-semming") to components of MWEs if special measures were not taken: (1) features of null-semmed verbs would be lost and (2) modifiers of null-semmed arguments would not have a head to modify. Since these problems require different solutions, we discuss them separately.

5.1 Recovering Features of Null-Semmed Verbs

In most verbal entries, whether or not the entry is a MWE, the verb is the head word, $var0, and there is never any reason (nor is there any method) to null-sem it because the whole entry is describing its meaning in the given configuration. As such, verbal features like tense, mood and aspect are always available to the semantic analyzer, as they should be. One such example is be-v7, which is headed by the verb *be*.

```
(be-v7
    (def "phrasal: X [be-pres] [xcomp] – indicates that X is going to happen in the
            future, it is planned")
    (ex "The president is/was to meet with the delegates in the lobby."))
    (syn-struc
        ((subject ((root $var1) (cat n)))
         (root $var0) (cat v)
         (xcomp ((root $var2) (cat v)))))
    (output-syntax cl)
    (sem-struc
        (^$var2
            (AGENT (value ^$var1))
            (time (> (find-anchor-time))))))
```

The semantic interpretation is: the event indicated in the complement clause (e.g., *meet with the delegates in the lobby*) will occur at a time after the anchor time; and the meaning of the syntactic subject (e.g., *the president*) is the agent of that event.

Contrast this with situations in which the feature-carrying verb is not the head word, as is the case in the sense for in-prep18 (the same is true for expect-v4, above). Here, the feature-carrying verb is *be*, which is our shorthand for “the verb *be* in any inflectional form or realized by a modal (e.g., *seem, appear*), or in combination with any modal (e.g., *seem to be, appear to be*).”

```
(in-prep18
   (def "phrasal: ‘X *be* in surgery’ = X *be* the EXPERIENCER of PERFORM-
         SURGERY")
   (ex "John was in surgery for 4 hours")
   (syn-struc
      ((subject ((root $var1) (cat n)))
       (v ((root $var2) (cat v) (root *be*)))
       (pp
          ((root $var0) (cat prep)
           (obj ((root $var3) (cat np) (root surgery)))))))
   (sem-struc
      (refsem1
         (PERFORM-SURGERY
            (EXPERIENCER (value ^$var1))))
      (^$var3 (null-sem +))
      (^$var2 (null-sem +))))
   (meaning-procedure
      (apply-meaning (strip-features (value ^$var2)) (value refsem1)))))
```

The entry says that *X {be} in surgery* means that X is the EXPERIENCER of the event PERFORM-SURGERY. As concerns the meaning of *be* ($var1), we need to null-sem it so that the semantic analyzer does not attempt to reanalyze this input using one of the productive meanings of *be* as the head. However, if we null-sem it outright, we will lose its semantic features as well, like tense, aspect and mood, even though they need to be applied to the semantic head of the proposition. Therefore, after we have null-semmed *be*, which is a semantic necessity, we strip the features from it and apply them to the semantic head – all of which is carried out using the compound meaning procedure shown above.

5.2 Recovering the meaning of Modifiers of Null-semmed Arguments and Adjuncts

An interesting issue arises when a component of an MWE whose meaning is non-compositional (e.g., *bucket* in *kick the bucket*) is modified (e.g., *kick the bloody bucket*). Such modifiers do not modify their syntactic heads, since those heads do not contribute compositionally to the whole. Instead, they bear *some* semantic relationship to the entire MWE – a relationship that must be determined using semantic analysis.

The problem of how modifiers contribute to proposition meaning is not specific to

MWEs, it is well known from the literature on adjectives. Consider the following three examples, drawn from Raskin and Nirenburg (1998), who in turn reference Vendler (1963, 1968):

- a beautiful dancer can be a beautiful *woman* who dances or a woman who *dances* beautifully
- a comfortable chair is a chair that *people feel* comfortable sitting in
- a fast car is a car that has the potential to *go* fast

In short, there is no lockstep correlation between syntactic structure and semantic interpretation, and robust natural language processing systems must be armed to reason about semantics on the fly.

There are two aspects of analyzing modified MWEs: syntactic analysis (parsing) and semantic analysis. The parsing of MWEs with modifiers poses no special problems since basic syntactic analysis permits the free inclusion of modifiers in any of the legal places for English. For example, the syntactic description of the MWE *kick the bucket* indicates that this is a transitive clause, that the root of the direct object must be *bucket* in the singular, and that the direct object must additionally contain the article *the* (the "contains" keyword is used to specify components of a category with no implication that the list is exclusive). There is nothing to block the free addition of modifiers either at level of clause or at the level of arguments – all clauses can take modifiers, as can all noun phrases.

```
(kick-v2
   (def "phrasal: kick the bucket = die")
   (ex “Last week his uncle kicked the bucket.”)
   (syn-struc
      ((subject ((root $var1) (cat n)))
       (root $var0) (cat v)
       (directobject
          ((root $var4) (cat n) (root bucket) (number sing)
           (contains
              (art ((root $var2) (cat art) (root the))))
    (sem-struc
      (DIE
         (EXPERIENCER (value ^$var1)))
      (^$var2 (null-sem +))
      (^$var4 (null-sem +)))
```

In short, free modification of MWEs is provided for without the need for any special expressive means.

As concerns semantics, our basic approach to treating modifiers within MWEs is to analyze the MWE as indicated in the sem-struc, then attempt to attach the meaning of unaccounted-for modifiers – which were syntactically hosted by null-semmed elements – to the meaning of the entire structure using generalized processes for meaning composition. Note that such meaning composition is not specific to MWEs: the analyzer carries out the same process in all cases when meaning must be recovered from an incomplete parse. The latter may be due to insufficient coverage of the parser, lexical lacunae that confound the parser, or unexpected (ungrammatical, highly elliptical, etc.) input.

Consider the example *He kicked the bloody bucket*. This sentence is ambiguous, with both a literal and a figurative reading possible: the man in question could have kicked a bucket covered in blood, or he could have died, with the speaker expressing this fact emphatically. As with all disambiguation in OntoSem, this disambiguation is supported by heuristics. If there is no coreferential category for *bucket* in the preceding context to explain the use of the definite article, then the idiomatic sense is preferred.[10] The TMR we would want to generate from this sentence is:

```
DIE-4
    EXPERIENCER   HUMAN-109
    EMPHASIS      .7  ;; relatively high emphasis - a style marker
    time          (< find-anchor-time)
```

The analyzer produces the basic semantics (DIE (EXPERIENCER HUMAN)) using the MWE entry in kick-v2, but it needs to productively add the meaning of emphasis, from *bloody*, to it. The first step is for the analyzer to look up *bloody* in the lexicon and determine which of its meanings would apply well to any of the elements of nascent TMR. The meanings of *bloody* are:

```
(bloody-adj1
    (def “related to blood”)
    (ex “That’s awfully bloody meat”)
    (syn-struc
        ((mods ((root $var0) (cat adj)))
         (root $var1) (cat n)))
  (sem-struc
    (^$var1
        (RELATION BLOOD))))

(bloody-adj2
    (def “indicates emphasis”)
    (ex “He’s a bloody fool!”)
    (syn-struc
        ((mods ((root $var0) (cat adj)))
         (root $var1) (cat n)))
  (sem-struc
    (^$var1
```

[10] Actually, there are other ways to explain a definite description with no coreferent: the given NP could be always definite (*the sun*) or could be used in a script that assumes the object to exist (*When milking a cow, first you take the bucket and put it under the cow*). Such heuristics are also included in our disambiguation engine.

```
    (EMPHASIS .7)))
```

The first sense is the literal one: as is necessary in computational lexicons, we do not split similar senses (dictionary.com has 3 senses for our 1) because doing so makes automated disambiguation more difficult, as was learned when machine-readable lexicons were applied to NLP. The second sense conveys emphasis. The analyzer must decide which meaning to apply to which element of the nascent TMR. There are 4 possibilities, which can be described informally as: the dying is literally bloody; the experiencer of it is literally bloody; emphasis is applied to the dying; emphasis is applied to the experiencer of it. In the abstract, this decision would be difficult to make; however, the analyzer has additional evidence: it knows that this modifier is an unanchored modifier in a MWE. As such, preference is given to modifier meanings that express emphasis or speaker attitude and scope over the entire proposition. Including such preferences in analysis algorithms, even if they are defeasible, is essential in order to cut through what would otherwise be potentially unresolvable ambiguity.

This first example was particularly complex, and was selected to show the real challenges facing text analyzers, even when they are armed with rich lexical and ontological knowledge. Most examples are not as complex. Consider the sentence *He attempted goddamned suicide*, for which we would want the analyzer to produce the following TMR:

```
KILL-6
      AGENT          HUMAN-110               ;; the agent and theme are
      THEME          HUMAN-110               ;; coreferential
      MODALITY-1     TYPE     EFFORT
                     VALUE    > .6           ;; he tried
      MODALITY-2     TYPE     EPISTEMIC
                     VALUE    < 1             ;; he didn't succeed
      MODALITY-3     TYPE     EVALUATIVE
                     VALUE    < .2           ;; the speaker isn't happy about it
      time           (< find-anchor-time)
```

The MWE for *X attempts suicide* and the single lexical entry for *goddamned* are as follows:

```
(attempt-v3
   (def "phrasal: attempt suicide")
   (ex "He attempted suicide.")
   (syn-struc
      ((subject ((root $var1) (cat n)))
       (root $var0) (cat v)
       (directobject ((root $var2) (cat n) (root suicide))))
   (sem-struc
      (KILL
         (AGENT (value ^$var1))
         (THEME (value ^$var1))
```

```
            (effort (> 0.6))
            (epistemic (< 1))
        (^$var2 (null-sem +)))))

(goddamned-adj1
    (def "indicates low evaluation by the speaker")
    (ex "His goddamned car broke down again")
    (syn-struc
        ((mods ((root $var0) (cat adj)))
         (root $var1) (cat n)))
 (sem-struc
    (^$var1
        (EVALUATIVE (< .2)))))
```

The processing of the unattached modifier (the modifier of the null-semmed *suicide*) in this example is much simpler than in the previous example because there is only one sense of *goddamned*. Following the general preference to apply unattached modifiers to the semantic heads of propositions rather than arguments, the low evaluative modality gets attributed to the entire proposition headed by KILL, as it should be.

All of our examples of modification of null-semmed components of MWEs involve emphasis. Indeed, this appears to be the main function of such modifiers since, by definition, modifying a null-semmed element cannot add literal meaning to that element. Of course, other types of modification can be found, as in *He kicked the proverbial bucket* and *He went and kicked the bucket*. These will either need to be incorporated into the static knowledge sources explicitly, processed dynamically at run time, or – best of all – folded into more general types of reasoning rules once a larger corpus of types of MWE modification has been compiled and studied.

6. To Create MWEs or Not To Create Them?

As stated in the introduction, deciding whether to lexicalize a MWE or to permit its runtime composition involves various considerations. Our practical work has led us to believe that there is no hard line between what should and should not be recorded as an MWE, and MWEs are not restricted to fully non-compositional collocations: they can be recorded because of their frequency, the high percentage of times that they are met with in a canonical form, etc. Consider entities like

- X {provide} services (to/for Y)
- X {cast} {spell} (over/on Y)
- X {emerge} from bankruptcy

These are very often used in precisely the form listed, with no added modifiers. Moreover, their semantic descriptions collapse the meaning of the whole in notable ways, as shown in the respective MWE senses.

When X provides services to Y, X is the AGENT of a SERVICE-EVENT whose BENEFICIARY is Y (provide-v2).

```
(provide-v2
    (def "phrasal: provide services = be the agent of a service event")
     (ex "This company provides services to elderly residents")
     (syn-struc
        ((subject ((root $var1) (cat n)))
          (root $var0) (cat v)
           (directobject ((root $var2) (cat n) (root service) (number pl)))
          (pp ((root $var3) (cat prep) (root (or to for)) (opt +)
               (obj ((root $var4) (cat n)))))
    (sem-struc
        (SERVICE-EVENT
            (AGENT (value ^$var1))
            (BENEFICIARY (value ^$var4)))
         (^$var2 (null-sem +)))
        (^$var3 (null-sem +))))
```

When X casts a spell over Y, X is the AGENT of a BEWITCH event whose BENEFICIARY is Y (cast-v1).

```
(cast-v1
    (def "phrasal: cast a spell over/on")
    (ex "She cast a spell over the mean cab driver")
    (syn-struc
        ((subject ((root $var1) (cat n)))
         (root $var0) (cat v)
         (directobject ((root $var2) (cat n) (root spell)))
        (pp
            ((root $var3) (cat prep) (root (or over on)) (opt +)
             (obj ((root $var4) (cat n)))))))
    (sem-struc
        (BEWITCH
            (AGENT (value ^$var1))
            (THEME (value ^$var4)))
        (^$var2 (null-sem +))
        (^$var3 (null-sem +))))
```

When X emerges from bankruptcy, X is the THEME of a BANKRUPT event whose phase is “end”.

```
(emerge-v2
    (def "in phrasal 'emerge (from bankruptcy)'"))
    (syn-struc
        ((subject ((root $var1) (cat n)))
       (root $var0) (cat v)
       (pp ((root from) (root $var2) (cat prep)
              (obj ((root $var3) (cat n) (root bankruptcy)))))))
 (sem-struc
     (BANKRUPTCY
         (THEME (value ^$var1))
         (phase end))
     (^$var2 (null-sem +))
     (^$var3 (null-sem +))))
```

To reiterate, we could rely on compositional analysis for these, but as with all cases of compositional analysis, we are opening the door to ambiguity since there are many productive senses of *provide, cast* and *emerge* in the lexicon. Therefore, there is something to be gained by lexicalizing particularly frequent phrasals and circumventing the need for disambiguation.

However, there is a price: the semantic descriptions of such entities include the null-semming of key components: in provide-v2, *service* is null-semmed; in cast-v1, *spell* is null-semmed, and in emerge-v2, *bankruptcy* is null-semmed. Obviously, we do not lose their meanings, they are folded into the meaning of the head of the sem-struc. However, if any of these words is modified in the textual input, the modifier becomes unbound – not unlike the case of *bloody* in *kick the bloody bucket.*

The problem with modifiers in less idiomatic MWEs is that they may or may not apply to the semantic head of the proposition. (Recall that we are hypothesizing that modifiers in truly idiomatic MWEs tend to apply to the head of the proposition.) Let us compare a few sentences with modified MWE elements and the TMRs we would want to be produced for them, then work back to the knowledge needed to automatically produce those TMRs.

They provide shopping services to elderly residents.

```
SHOPPING-1                                  ;; provide shopping services
    AGENT          SET-4                    ;; they
    BENEFICIARY    HUMAN-45                 ;; elderly residents
```

This TMR captures the fact that *provide shopping services for* means the same thing as *shop for*. Every language provides many such equivalent collocations, including *the month of January* (mentioned above), *the city of London*, etc. We could prepare the analysis system to generate such a TMR by expanding our description of provide-v2 as follows (the additions are in bold face):

```
(provide-v2
    (def "phrasal: provide services = be the agent of a service event")
     (ex "This company provides services to elderly residents")
     (syn-struc
         ((subject ((root $var1) (cat n)))
           (root $var0) (cat v)
```

```
      (directobject ((root $var2) (cat n) (root service) (number pl)
                    (contains (n ((root $var5) (cat n) (opt +))))))
     (pp ((root $var3) (cat prep) (root (or to for)) (opt +)
          (obj ((root $var4) (cat n)))))
  (sem-struc
     (refsem1
        (SERVICE-EVENT
           (AGENT (value ^$var1))
           (BENEFICIARY (value ^$var4))))
     (^$var2 (null-sem +)))
     (^$var3 (null-sem +))))
  (meaning-procedure
     (if ^$var5 (sem SERVICE-EVENT)
     then replace (refsem1.head) (value ^$var5))))
```

In the sem-struc, we have explicitly stated that the direct object might optionally contain a non-head nominal (i.e., creating a n-n compound), and if it does, it will be assigned the variable name $var5. We need that variable name as a parameter for the added meaning procedure, which says: "If the meaning of $var5 is of the semantic type SERVICE-EVENT, then replace the SERVICE-EVENT that heads the sem-struc with this EVENT." Nominal modifiers with other meanings – as in *provide valet services* – will be treated in the usual way, as unbound modifiers that need to be linked to the most semantically appropriate element of the TMR.

Explicitly encoding such expectations takes time and the list will never be truly complete, so having more generalized strategies is essential. Consider what would happen if we permitted the analyzer to process *They provide shopping services to elderly residents* with our old, more generalized version of provide-v2. The unbound modifier *shopping* would be added to the TMR using the generalized RELATION, as follows:

```
SERVICE-EVENT-1                      ;; provide shopping services
   AGENT          SET-4              ;; they
   BENEFICIARY    HUMAN-45           ;; elderly residents
   RELATION       SHOPPING
```

This result is actually not bad: it says "there was a service event related to shopping…," which is slightly less specific than the ideal version but still contains all the needed concepts linked by correct properties.

In fact, in many cases, free-form semantic composition works quite well. Temporal modifiers are a good case in point, as shown by the example *The company emerged from a lengthy bankruptcy*. Using just our basic MWE sense, emerge-v2, the preliminary TMR would be:

```
BANKRUPTCY-1
   THEME          CORPORATION-12
   phase          end
```

```
    time            (< find-anchor-time)   ;; due to the past tense verb
```

There is only one sense of *lengthy* in our lexicon, which modifies an event and adds to it (DURATION (> .8)). In this TMR, there is only one event that *lengthy* could modify, BANKRUPTCY, so that is what it modifies, yielding the complete TMR:

```
BANKRUPTCY-1
    THEME           CORPORATION-12
    phase           end
    time            (< find-anchor-time)
    DURATION        > .8
```

Given sufficient time, one could specify quite precisely how to deal with different classes of modifiers in MWEs. These specifications would be written in meaning procedures using the standard knowledge representation means, reference to ontological subtrees, etc. For example, a typical modifier that could be added to the quasi-idiomatic *cast a spell on someone* is a description of what kind of spell: evil, deadly, wicked, personality-altering… During acquisition, one could choose to make the semantic description a bit redundant and, instead of null-semming *spell*, make it the INSTRUMENT of BEWITCH.

```
(cast-v1
    (def "phrasal: cast a spell over/on")
    (ex "She cast a spell over the mean cab driver")
    (syn-struc
        ((subject ((root $var1) (cat n)))
         (root $var0) (cat v)
         (directobject ((root $var2) (cat n) (root spell)))
        (pp
            ((root $var3) (cat prep) (root (or over on)) (opt +)
             (obj ((root $var4) (cat n)))))))
    (sem-struc
        (BEWITCH
            (AGENT (value ^$var1))
            (THEME (value ^$var4))
            (INSTRUMENT (value ^$var2)))
        (^$var3 (null-sem +))))
```

While this might look like overkill for non-modified inputs, it is exactly what is needed for modified ones: leaving ^$var2, *spell*, as a full-fledged component of the sem-struc means that any modification to it will be carried out in the usual way.

The goal of this discussion has been to lay out the choice space for acquirers when faced with a multiplicity of competing demands when building resources for high-end NLP. While lexicalizing MWEs can help in ambiguity resolution and can be a quick fix for immediate processing needs, it can have undesirable consequences for the processing of free modification. And while those consequences can be moderated by encoding more expectations in the MWE entry, that process can be never-ending and therefore practically unrealistic. Keeping the big picture in mind at all times is important. For example, rather than encode MWEs for *have a smoke, have a drink, have a snack*, one can create a generalized MWE composed of *X {have} {NP},* in which the NP is semantically constrained to an EVENT; the semantic interpretation is, then, that X is the AGENT of that EVENT.

7. Application Areas

The decisions of what to pursue in NLP and how to pursue it depend in large part on applications. As noted earlier, simply detecting MWEs is sufficient for some applications, whereas for others a full understanding of their semantics is required.

Our environment, OntoSem, is used for both broad-domain and narrow-domain applications. An example of a recent broad-domain application is automatically creating TMRs of texts taken from different subject domains with an emphasis on the processing of modality. An example of a current narrow-domain application is dialog support for the Maryland Virtual Patient (MVP) project. MVP is an agent-oriented simulation and tutoring system in which a human user plays the role of a physician in training who must diagnose and treat open-ended simulations of patients, with or without the help of a virtual mentor agent (see, e.g., McShane et al. 2007). The virtual patient is, itself, a “double” agent, comprised of a physiological agent that lives over time and responds in realistic ways to disease progression and interventions, and a cognitive agent that experiences symptoms, decides when to consult a physician, makes decisions about its lifestyle, treatment options, etc., and communicates with a human user using natural language. All of the communication in MVP must be maximally precise, not permitting either precision or recall to be low.

8. Final Thoughts

Multi-word expressions present an interesting combination of fixed and mutable aspects. In order for an NLP-oriented lexicon to robustly treat MWEs, it must provide for great variation in MWEs, both syntactically and semantically. Some such variation can be anticipated and folded into knowledge acquisition, while other aspects must be handled at runtime, most profitably by semantically-grounded reasoning. Although MWEs present some special challenges, most of them can be met within a semantically oriented text processing environment using the same basic methodologies used for all other aspects of text processing: a fully integrated combination of syntactic and semantic analysis coupled with procedural semantic routines launched at runtime Therefore, while we appreciate that MWEs can be considered “a pain in the neck” for NLP (Sag et al. 2003), within our environment they do not cost more acquisition time than other lexical items.

References

Baldwin, Timothy and Aline Villavicencio. 2002. Extracting the Unextractable: A Case Study on Verb-particles, In Proceedings of the Sixth Conference on Computational Natural Language Learning (CoNLL 2002), Taipei, Taiwan, pp. 98-104.

Beale, S., S. Nirenburg and K. Mahesh. 1995. Semantic analysis in the Mikrokosmos machine translation project. Proceedings of the 2nd Symposium on Natural Language Processing, Bangkok, Thailand.

Beale, S., S. Nirenburg and M. McShane. 2003. Just-in-Time grammar. Proceedings 2003 International Multiconference in Computer Science and Computer Engineering. Las Vegas, Nevada.

Cacciari, Cristina and Patrizia Tabossi. 1993. Idioms: Processing, Structure and Interpretation. Lawrence Erlbaum and Associates, Inc.

Calzolari, Nicoletta, Charles Fillmore, Ralph Grishman, Nancy Ide, Alessandro Lenci, Catherine MacLeod, and Antonio Zampolli. 2002. Towards best practice for multiword expressions in computational lexicons. In Proc. of the 3rd International Conference on Language Resources and Evaluation (LREC 2002), pages 1934–40, Las Palmas, Canary Islands.

Dias, Gaël . 2003. Multiword unit hybrid extraction. Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment, pp. 41 – 48, Sapporo, Japan.

Fellbaum, C. 1998. Towards a Representation of Idioms in WordNet. Col-WordNet, pp. 52-57.

Harabagiu, S.; Miller, G.; and Moldovan, D. 1999. WordNet 2 - A Morphologically and Semantically Enhanced Resource. In Proceedings of SIGLEX-99. pp.1--8. University of Maryland.

Ide, N. and J. Véronis. 1993. Extracting knowledge bases from machine-readable dictionaries: Have we wasted our time? Proceedings of KB&KS'93 Workshop. Tokyo, pp. 257-266.

Inkpen, Diana Zaiu and Graeme Hirst. 2001. Experiments on extracting knowledge from a machine-readable dictionary of synonym differences." In: Gelbukh, Alexander (editor), Computational Linguistics and Intelligent Text Processing (Proceedings, Second Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, February 2001), Lecture Notes in Computer Science 2004, Berlin: Springer-Verlag, pp. 264-278.

Kim, Su Nam and Timothy Baldwin (2006) Automatic Identification of English Verb Particle Constructions using Linguistic Features, In Proceedings of the Third ACL-SIGSEM Workshop on Prepositions, Trento, Italy, pp. 65–72.

Lubensky, S. [1995]. A Russian-English Dictionary of Idioms. Random House.

McShane, M., S. Beale, and S. Nirenburg. 2004. Some meaning procedures of Ontological Semantics. Proceedings of LREC-2004, Lisbon, Portugal.

McShane, M., S. Nirenburg, S. Beale, B. Jarrell, G. Fantry. 2007. Knowledge-based modeling and simulation of diseases with highly differentiated clinical manifestations. Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME 07), Amsterdam, The Netherlands, July 7-11, 2007.

Nirenburg, S., M. McShane and S. Beale. 2003. Operative strategies in Ontological Semantics. Proceedings of the HLT-NAACL-03 Workshop on Text Meaning, Edmonton, Alberta, Canada.

Nirenburg, S., M. McShane and S. Beale. 2005. Increasing understanding: Interpreting events of change. Proceedings of the OntoLex Workshop at IJCNLP-05, Jeju Island, South Korea.

Nirenburg, Sergei and Victor Raskin. 2004. Ontological Semantics. The MIT Press.

O'Hara, T.; and Wiebe, J. 2003. Preposition Semantic Classification via Treebank and FrameNet. In Proceedings of the Conference on Natural Language Learning (CoNLL-03), Edmonton.

Raskin, V., and S. Nirenburg. 1998. An applied ontological semantic microtheory of adjectival meaning for natural language processing. Machine Translation 13(2-3):135-227.

Sag, I., T. Baldwin, F. Bond, A. Copestake and D. Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2002), Mexico City, Mexico, pp. 1-15.

Schone, Patrick and Daniel Jurafsky. 2001. Is Knowledge-Free Induction of Multiword Unit Dictionary Headwords a Solved Problem? Proceedings of Empirical Methods in Natural Language Processing, Pittsburgh, PA.

Sharoff, Serge. 2004. What is at Stake: a Case Study of Russian Expressions Starting with a Preposition. ACL 2004 Workshop

Smadja, Frank. 1993. Retrieving collocations from text: Xtract. Computational Linguistics Volume 19 , Issue 1, Pages: 143 - 177

Stock, Oliviero, Jon Slack and Andrew Ortony. 1993. Building castles in the air: Some computational and theoretical issues in idiom comprehension, in Cacciari and Tabossi 1993, pp. 229-248.

Vendler, Zeno 1963. The grammar of goodness. The Philosophical Review 72:4, pp. 446-465.

Vendler, Zeno 1968. Adjectives and Nominalization. The Hague: Mouton.

Venkatsubramanyan, Shailaja and Jose Perez-Carballo. 2004. Multiword Expression Filtering for Building Knowledge. ACL 2004 Workshop.