

Semi-automatic verb-driven lexicon acquisition enhancer

Max Petrenko
Princess Ekaterina R. Dashkova Moscow Humanities Institute
and RiverGlass, Inc.
Moscow, Russia, and Champaign, IL, USA
maxpetrenko@alumni.purdue.edu

Abstract – *the paper discusses ways of enhancing lexical acquisition within Ontological Semantic Technology without the detriment to the grain size of lexical coverage. The semi-automatic verb driven enhancer is introduced which derives nominal and adjectival senses from manually acquired verb senses by converting their case role fillers into complete sense templates. The declarative and procedural components of the enhancer are described, and its functionality is demonstrated.*

Keywords: computational lexicon, lexicon acquisition, language engineering, ontological semantics

1 Introduction

The need and rationale for developing an incremental utility for enhancing lexicon acquisition emerged out of several years of semi-automatic acquisition of the English lexicon within the framework of Ontological Semantic Technology (OST) – an already well-established and fully functional system featuring a full-fledged methodology and toolbox for knowledge acquisition. While the automation of lexicon acquisition in OST is already achieved through format consistency check in lexical senses and ontological fillers validity check in semantic structures, further possibilities have become available, mostly prompted by the richness of the increasingly robust ontology. From the enhancer’s adherence to the OST framework, two premises follow. First, the declarative and procedural components of the enhancer fully reflect the premises, methodology of lexicon acquisition postulated in [3] and further discussed in [6]. Second, the introduced utility aims at enhancing, rather than supplanting, the efforts of a human acquirer, and is meant to further address the common charges of time-inefficiency of acquisition efforts and the incompleteness and non-uniformity of resources traditionally leveled against meaning-driven human-mediated lexicon acquisition [4]. The latter premise

also implies that the incremental application of the enhancer does not warrant its systemic comparison to large-scale fully automatic systems like [5].

The paper will briefly outline the framework of OST, discussing pertinent aspects and lexicon acquisition strategies and techniques discussed in [3] and [6]. The semi-automatic verb-driven enhancer will then be introduced in detail. The enhancer’s functionality and efficiency will then be illustrated with a detailed step-by-step description of senses extraction from a verb sense.

1.1 Knowledge resources of OST

As argued, illustrated and practically demonstrated in an increasingly growing number of publications [1, 7, 8, and 3], OST is predicated on the premise that scalable computation of natural language text requires a robust, descriptively rich and reusable knowledge resource platform on the basis of which natural language meaning could be accessed directly rather than being guessed via collocational or syntactic dependencies [7].

The knowledge resource platform comprises a language-independent ontology – a lattice of concepts defined through a rich network of properties – and a language-specific lexicon, capturing word senses including orthographic, syntactic alternations, incorporating optional prepositional or phrasal adjuncts, and representing non-compositional units.

The architecture, axiomatic description and concept structure has been extensively discussed in [4, 8]. Stemming from the top-level OBJECT-EVENT-PROPERTY branching, OBJECT and EVENT concepts are defined via an extensive set of PROPERTY concepts in the (concept(property(facet(filler)))) format. Inheritance and transitivity relations are enforced. Filler preference is captured via facets SEM and DEFAULT, and numeric value ranges of ATTRIBUTES can be represented via RELAXABLE-TO.

Lexicon entries relate natural language syntax and semantics by co-indexing syntactic positions (subject,

direct object, etc.) with semantic property fillers (CASE-ROLE's for EVENT senses, RELATION's for OBJECT senses, and DOMAIN/RANGE for ATTRIBUTE senses). A lexical entry template is shown below:

```
(head-entry
  (sense-1, 2, 3...
    (cat(n/v/adj/pro/prep))
    (synonyms "")
    (anno
      (def "")
      (comments "acquisition time stamp or other
notes")
      (ex ""))
    )
  (syn-struct(
    (root($var0))(cat(n/v/adj/pro/prep))
    (subject/object((root($var#))(cat(np/vp/s))))
  )
  (sem-struct
    (root-concept
      (property(value(^$var#
(restricted-to(default/sem(concept)))))))
  )
  )
)
```

During computation, the text processor will derive a text meaning representation (TMR) of a sentence by mapping syntactic positions onto semantic roles, in the order stipulated by the syntactic structure (“syn-struct”) of the sense, where \$var0 stands for the word sense per se, and all additional variables indicate adjacent optional or obligatory arguments. Order variation is accommodated through additional syntactic structures. The semantic structure (“sem-struct”) defines the meaning of the sense by anchoring it in the most representative concept (i.e. root-concept) adding properties where necessary. Restriction of property filler variables is often required because anchor concepts may be too generic to capture a particular sense. An example sense of “ax-v1” is listed below:

```
(ax-v1
  (cat(v))
  (anno(def "to destroy or break with an
ax")(comments ""))
  (ex "he axed the door to break out of the burning
house")(synonyms "")
  (syn-struct(
    (subject((root($var1))(cat(np))))
    (root($var0))(cat(v))
    (directobject((root($var2))(cat(np))))
    (sem-struct(destroy(agent(value(^$var1(restricted-
to(human)))))(instrument(sem(ax)))
    (theme(value(^$var2(restricted-to(sem(plant
artifact))))))))))
```

Since DESTROY and its case roles of AGENT, INSTRUMENT and THEME are too general to capture the meaning of “ax-v1”, case role fillers are “tightened”: from ANIMAL to HUMAN (for the AGENT case role), from TOOL to AX (for the INSTRUMENT case role), and from PHYSICAL-OBJECT to PLANT ARTIFACT (for the THEME case role).

1.2 Lexicon acquisition in Ontological Semantics

In OST, lexicon acquisition is guided by the following premises [see 3 for a detailed discussion]:

- 1) Lexicon coverage must be complete, i.e. represent all senses, for a target domain;
- 2) The breadth and depth (i.e. “grain size”) of lexicon acquisition is determined by application objectives;
- 3) Maintaining a determined grain size is crucial for text meaning representation and overrides considerations of time efficiency;
- 4) Automatic (i.e. “runtime”) rule-based sense acquisition has no decisive advantage over semi-automatic rule-based sense acquisition in OST because the generative capacity of acquisition rules is unaffected by whether they are triggered at runtime or acquisition time.
- 5) Human involvement in OST lexicon acquisition is instrumental in aspects where semantic competence is needed: selecting an appropriate head concept, determining the optionality of syntactic elements, defining the number of senses per entry, reducing polysemy and post-acquisition testing.
- 6) The automation of lexicon acquisition in OST should be incremental, i.e. used sparingly so as to avoid supplanting or impeding human involvement in semantic aspects outlined in 5.

Rule-based and propagation-based strategies of lexicon acquisition have been described in [3], and ontology-driven, parser-driven or domain-driven acquisition techniques outlined in [6].

2 Enhancer description

The section introduces the declarative (goals, resource components, input/output specifications) and procedural (rules and derivation phases) components of the semi-automatic verb-driven lexicon acquisition enhancer.

2.1 Declarative component of the enhancer

The enhancer generates nominal and adjectival senses from a manually acquired verb sense. The generation involves “unpacking” case role fillers (obtainable from case role variable restrictions or ontological case role fillers) of the verb sense into respective noun senses, or syn-struc and sem-struc modifications of the verb sense in order to derive deverbal nouns or adjectives. Inverse property calculation is crucial for the derivation. Once a complete lexical sense (including the annotation, syn-struc and sem-struc fields) has been created, an appropriate name is looked up and, upon the approval of a trained acquirer, assigned to the sense. Once a sense has been named and confirmed by the acquirer, an appropriate synonym list is automatically retrieved for the given name, and the sense’s complete template is propagated throughout the synonym list. The procedure is semi-automatic in that the enhancer will first attempt to derive a sense automatically and will request acquirer’s approval at every phase of the process.

The enhancer builds on the already existing lexicon acquisition technology described in [3, 6]. In order to increase the breadth of coverage for the enhancer output, a repository of noun and adjective synonym lists is included as a separate resource component. The synonym lists are formalism-free, i.e. consist of simple words without definitions, notational encoding or ordering preferences:

<house dwelling homestead home shack ...>
<workable doable practical feasible applicable...>

What matters for the enhancer is a paradigmatic lexical set with predetermined semantic similarity, which will receive full interpretation upon the completion of the enhancement procedure.

Two types of verb senses serve as input to the enhancer: verbs with unrestricted case role variables; and verbs with restricted case role variables.

An unrestricted case role variable implies that the case role filler is specified in the ontology and no tightening is necessary for the sense. Consider the sem-struc of “travel-v1”:

```
(travel-v1
  (cat(v))
  (anno(def "to move via transport") (ex "He
traveled from France to Spain. The ship traveled to
South Africa"))
  <...>
  (syn-struc((subject((root($var1))(cat(np))))
    (root($var0))(cat(v))))
  (sem-struc(travel(agent(value(^$var1)))
    (instrument(value(^$var1))))))
)
```

Both AGENT and INSTRUMENT case roles are sufficiently constrained for the concept TRAVEL in the ontology:

```
(travel
  (definition (value("to move via transport")))
  (is-a (hier(change-location culture-event)))
  (subclasses(hier(drive)))
  (agent(sem(human)))
  (instrument(sem(vehicle)))
  <...>
)
```

From the enhancer perspective, deriving a sense from an unrestricted case role variable means retrieving the sem-struc head concept from the respective ontological filler of the given case role.

A restricted case role variable indicates that the case role filler specification in the ontology is too generic for the given verb sense and needs tightening. Consider a verb sense of “board”:

```
(board-v1
  (cat(v))
  (anno(def "to move on board of a vehicle")
    (ex "the passengers boarded the plane"))
  (syn-struc((subject((root($var1))(cat(np))))
    (root($var0))(cat(v))(directobject((root($var2))(cat(n
p))))))
  (sem-struc(walk(agent(value(^$var1)))
    (destination(value(^$var2(restricted-
to(sem(vehicle))))))))))
```

No directly appropriate concept for the event of boarding transport exists in the ontology, and the restrictions on the DESTINATION case role for the nearest available concept WALK are too general to capture the meaning of “board-v1”, therefore the DESTINATION filler is “tightened” from PHYSICAL-OBJECT to VEHICLE. From the enhancer perspective, a case role variable restriction often contains, other conditions being met, an explicitly defined sem-struc head of the new entry.

The enhancer generates the following types of senses:

a) Case role-based noun senses are derived from case role fillers of the verb and are listed below by respective case roles in the ontology:

- agentival noun (AN): research-v → researcher-n, scholar-n, scientist-n, etc.
- instrumental noun (IN): eat-v → spoon-n, fork-n,
- thematic noun (TN): discard-v → trash-n, garbage-n, junk-n, etc.
- benefactive noun (BN): drive-v → passenger-n
- locative noun (LN) derived from case roles of LOCATION, PATH, SOURCE, DESTINATION: meal-v → restaurant-n; board-v → gangway-n; jump-v → springboard-n, etc.

b) Deverbal noun senses are derived by converting syntactic positions of the verb into optional

prepositional adjuncts without any modifications to the sem-struct: research-v → research-n; drive-v → drive-n, etc.

c) Adjectival senses are normally derived by modifying the syn-struct and sem-struct of the verb: work-v → workable-adj, eat-v → edible-adj

At the sense approval/revision phase, it is important for the acquirer to remember that the enhancer is aimed at generating only those nominal and adjectival senses whose semantics is directly informed by the case role properties used in the sem-struct of the verb. To illustrate, an agentive noun sense “teacher-n” can be derived from “teach-v1”. While the AGENT case role of the sem-struct of “teach-v” warrants any noun sense anchored in HUMAN (driver-n, brother-n, criminal-n, etc.), only “teacher-n1” will have its semantics exhaustively encapsulated by the (human(agent-of(default(teach)))) expression. Simply put, any human is capable of teaching, but for “teacher” the quality of being the agent of teaching defines the word’s existence. To establish the indispensability of a case role property for the meaning of a sense, a criterion can be applied: if eliminating the case role property from the sem-struct of the sense (e.g. “teacher-n”) makes it synonymous with its hypernym (e.g. “human-n”), then the case role property is semantically indispensable (a similar distinction between categorical semantic markers shared by all class members and sense-specific distinguishers delineating a specific sense was drawn in [2]).

2.2 Procedural component of the enhancer

The enhancer involves three procedures. The description below spares the reader many algorithmic details and rather aims to outline the general logic of the cycle. The three procedures are not linearly ordered and may be called recursively.

(1) Sense derivation

Case role noun derivation

The general rule involves creating an inverse of the respective case role in the sem-struct of the verb sense, so that (event(case-role(facet(filler)))) → (filler(case-role_inverse(facet(event)))).

For example, an agentive noun sense will have its sem-struct derived by replacing the AGENT case role with AGENT-OF and reversing its domain and range fillers so the original (event(agent(sem/default(filler)))) is rewritten as (filler(agent-of(sem/default(event)))). Other case role noun inverse derivations are listed below:

IN: (filler(instrument-of(sem/default(event))))
 TN: (filler(theme-of(sem/default(event))))
 BN: (filler(beneficiary-of(sem/default(event))))

LN: { (filler(location-of(sem/default(event))))
 (filler(path-of(sem/default(event))))
 (filler(source-of(sem/default(event))))
 (filler(destination-of(sem/default(event))))

If the verb sem-struct contains an unrestricted case role variable, the ontological filler is found under the respective EVENT: the (filler(case-role_inverse(sem/default(event)))) string is searched for in the definition of the respective EVENT case role filler or any of its children. If the search returns a concept, it becomes the head of the sem-struct the derived noun sense. Upon acquirer’s approval, the sense is created with the anno and syn-struct fields copied from the noun template and the found concept in the head of the sem-struct. The inverse property is omitted to avoid the reduplication of ontological properties:

```
(...-n1
  (cat(n))
  (synonyms "")
  (anno(def "")(comments "")(ex ""))
  (syn-struct((root($var0))(cat(n))))
  (sem-struct(filler))
)
```

If the verb sem-struct contains a restricted case role variable, the inverse expression is derived from the variable restriction based on the respective case role rule, and the (filler(case-role_inverse(sem/default(event)))) string becomes the sem-struct of the noun entry. A complete noun sense is created and offered for acquirer’s approval:

```
(...-n1
  (cat(n))
  (synonyms "")
  (anno(def "")(comments "")(ex ""))
  (syn-struct((root($var0))(cat(n))))
  (sem-struct(filler(case-
    role_inverse(sem/default(event))))))
)
```

Deverbal noun derivation

Deverbal nouns are derived syntactically, i.e. by replacing the syntactic roles of subject and object of the verb syn-struct with optional prepositional adjuncts. The conversion rules are listed below:

For subject:

(subject((root(\$var1))(cat(np)))) → (pp-adjunct((root(by))(opt(+))(cat(pre)))(obj((root(\$var1))(cat(np))))))

For direct object:

(directobject((root(\$var2))(cat(adj)))) → (pp-adjunct((root(of))(opt(+))(cat(pre)))(obj((root(\$var2))(cat(np))))))

The sem-struct of the verb sense remains unchanged. After the syn-struct conversion is complete, the sense is created:

```
(...-n1
  (cat(n))(synonyms "")
  (anno(def "an act of __") (ex ""))
  (syn-struct((root($var0))(cat(n))
    (pp-adjunct((root(of))(opt(+))
      (cat(pre)))(obj((root($var1))(cat(np))))))
    (pp-adjunct((root(by))(opt(+))
      (cat(pre)))(obj((root($var2))(cat(np))))))
    (sem-struct(copied from the verb sense)))
)
```

Adjective derivation

The class of deverbal adjectives is very narrow and restricted only to those modifying likely non-agentive case role fillers of the EVENT in the verb sem-struct. The derivation procedure involves creating a standard adjectival sense whose head sem-struct variable is defined through the inverse property of any case role filler other than agent indicated in the verb sense, and a high probabilistic modality value, e.g. (epistemic(value(0.8))) assigned to the EVENT:

```
(...-adj
  (cat(adj))
  (anno(def "")(comments "")(ex ""))
  (synonyms "")
  (syn-struct((root($var0))(cat(adj))
    (syn-np((root($var1))(cat(np))))))
  (syn-struct1((syn-np((root($var1))(cat(np))))
    (vp((root(be))(cat(v))))(adj-comp((root($var0))(cat(adj))))))
  (sem-struct(^$var1(case-
role_inverse(sem/default(event(epistemic(value(0.8))
))))))
)
```

Thus, while deriving deverbal adjectives, the enhancer will ignore the agent and try and convert any other case role filler, e.g.:

THEME (read-v → readable-adj)
 BENEFICIARY (adore-v → adorable-adj)
 INSTRUMENT (drive-v → drivable-adj)
 PATH (pass-v → passable-adj)

(2) Name assignment

The name assignment procedure determines whether the newly created sense skeleton, complete with syntactic and semantic specifications, is lexicalized in English, i.e. has a name and enters a synonym class. While automatic name search steps will first be undertaken, acquirer’s approval is often crucial at this phase. In order to establish lexicalization, the system will exploit the property of ontological concept labels which acquirers trained in

OST are normally instructed to be conscious of: the possible overlap of the concepts labels with language lexemes. This said, presuming that, while concepts and their labels are language independent, the choice of a concept label may have been motivated by a prototypical lexical token, the system will check whether the label of the sem-struct head concept matches any of the words available in the synonym list repository. If a match is found, the label will be suggested for the sense name, and acquirer’s approval will be requested. If no match is found, the acquirer will be requested to assign the name manually.

Manual name assignment is most often required in adjectival sense derivation for two reasons. First, most deverbal adjectives reluctantly form synonymous relations, which reduces the available synonym list repository thus preventing automatic search. Second, while deverbal adjectives of the type above at the first glance have a relatively uniformed morphological structure and seem to easily lend themselves to automation, a detailed look reveals a more complex picture. The suffixation process involves internal variation in the suffix (-able/ible) and the verb stem (e.g. final consonant reduplication in cases like “regrettable”) based on the morphophonemic properties of the stem, the surface phonetic representation of the final “e” (e.g. value → valuable, but foresee → foreseeable) – a task whose automation, if possible and justifiable, requires a separate research and development effort. Having the acquirer assign the name manually would seem a more time-efficient strategy in the current implementation.

The acquirer’s role at this phase is determining whether the created sense deserves a name, i.e. a word exists in English whose meaning is exhaustively encapsulated in the sem-struct of the created sense; the indispensability criterion as described in 2.1. is also made use of.

(3) Synonym check

Synonyms are assigned to the created sense by propagating its template among all members of the synonym list. Acquirer’s approval is required to establish the semantic grain size of the coverage, which will also determine the synonym class size.

3 Enhancer functionality

The current section illustrates how the semi-automatic verb-driven lexicon acquisition enhancer generates nominal and adjectival senses from the manually acquired verb sense of “drive-v”:

```
(drive-v1
  (cat(v))
  (anno(def "to change location by operating a
vehicle")
  (ex "he drove his car. he drove across the
city. tanks drove in the city")(synonyms ""))
  (syn-struct((subject((root($var1))(cat(np))))
(root($var0))(cat(v))(directobject((root($var2))(opt(+
))(cat(np))))))
  (syn-struct1((subject((root($var2))(cat(np))))
(root($var0))(cat(v)))
  (sem-struct(drive
(agent(value(^$var1))
(instrument(value(^$var2))))))
)
```

Deriving case role nouns from “drive-v”

Agentival nouns

Since the variable ^\$var1 on AGENT is unrestricted, ontological fillers of AGENT for DRIVE are checked, and the concept HUMAN is detected. The inverse of (drive(agent(sem(human))) is derived, and HUMAN and all its children concepts are checked for the expression (agent-of(sem(drive))) in their definition. After the automatic search returns no results, the acquirer is asked to manually add the head concept:

```
(-n1
  (cat(n))
  (synonyms "")
  (anno(def "agent of driving")(comments
""))(ex ""))
  (syn-struct((root($var0))(cat(n))))
  (sem-struct(human(agent-of(sem(drive))))))
)
```

The synonym list <human homo sapiens man> returned automatically for “human” by the name assignment procedure will be discarded because the (agent-of(sem(drive)) expression does not exhaustively encapsulate the meaning of the synonym list members. Thus, the acquirer will add the sense name manually:

```
(driver-n1
  (cat(n))
  (synonyms "")
  (anno(def "agent of driving")(comments
""))(ex "the driver stopped the car"))
  (syn-struct((root($var0))(cat(n))))
  (sem-struct(human(agent-
of(default(drive))))))
```

)

The synonym check procedure will search for a match of “driver” and return the list <automobilist, cabbie, chauffeur, motorist, operator>. Upon the acquirer’s approval, six new noun senses will be generated by propagation:
automobilist-n1, cabbie-n1, chauffeur-n1, morotist-n1, operator-n1

Instrumental nouns

Since the variable ^\$var2 on the INSTRUMENT is unrestricted, ontological fillers of the INSTRUMENT case role of the concept DRIVE are checked, and concepts CAR BUS TRACTOR and TRUCK are detected. The inverse of (drive(instrument(sem(car bus tractor truck military-tank)))) is derived, and each of the fillers and their children are checked for the expression (instrument-of(sem(drive))) in their definition. In the ontology where CAR, BUS, TRACTOR and TRUCK are subsumed under the same subclass LAND-DRIVEN-VEHICLE, the expression (instrument-of(default(drive))) justifiably contrasts it with siblings like AIR-VEHICLE, whose default INSTRUMENT-OF filler is FLY.

After (instrument-of(default(drive))) is found in the definition of CAR, BUS, TRACTOR and TRUCK, the enhancer will automatically create a sense template for each concept, removing duplicate ontological properties from their sem-structs, e.g.:

```
(-n1
  (cat(n))
  (synonyms "")
  (anno(def "instrument of
driving")(comments ""))(ex ""))
  (syn-struct((root($var0))(cat(n))))
  (sem-struct(car))
)
```

The name assignment procedure will establish the lexicalization for “car”, “bus”, “tractor” and “truck” by finding respective matches in the synonym list repository and automatically assigning names “car-n1”, “bus-n1”, “tractor-n1”, and “truck-n1” to sense templates with respective sem-structs.

After the acquirer’s confirmation, the synonym check procedure will return appropriate lists for each match, e.g.

“car-n1”: <auto, automobile, clunker, convertible, coupe, gas guzzler, hardtop, hatchback, jalopy, jeep, limousine, motor, motorcar, pickup, roadster, sedan, station wagon, subcompact, touring car>;

“bus-n1”: <coach, autobus, omnibus, passenger vehicle, public bus>

“tractor-n1”: <caterpillar>

“truck-n1”: <lorry, eighteen-wheeler, semi, pickup>

Noun senses (nineteen for “car-n”, six for “bus-n”, two for “tractor-n” five for “truck-n” and their synonyms) are then created by rapid propagation. The sense of “car-n1”, complete with synonyms, is listed below:

```
(car-n1
  (cat(n))
  (anno(def "a land driven vehicle")
  (synonyms " auto-n1, automobile-n1,
  clunker-n1, convertible-n1, coupe-n1, gas-
  guzzler-n1, hardtop-n1, hatchback-n1,
  jalopy-n1, jeep-n1, limousine-n1, motor-n1,
  motorcar-n1, pickup-n1, roadster-n1, sedan-
  n1, station-wagon-n1, subcompact-n1,
  touring-car-n1")
  (comments "")(ex "Bill sold his old car"))
  (syn-struct((root($var0))(cat(n))))
  (sem-struct(car))
)
```

Deverbal noun senses

For “drive-v1”, a deverbal noun “drive”, as in “he wrote a review of the test drive of the new model by an expert”, is available. First, derivation rules apply to the syn-struct of “drive-v1”, where the syntactic subject is re-written as an optional prepositional adjunct with “by”, and the direct object is re-written as an optional prepositional adjunct with “of”. The optionality of the prepositional adjuncts is dictated by cases like “he took his first drive to the countryside”, where both adjuncts are omitted. The sem-struct of “drive-v1” remains unchanged. The sense name is assigned automatically and approved by the acquirer, and a synonym list is retrieved: <ride, commute>. The complete sense is then derived:

```
(drive-n1
  (cat(n))
  (anno(def "an act of driving a land
  vehicle")(ex "he took a drive to the countryside. he
  reviewed the test drive of the new model by the
  expert")(comments ""))
  (synonyms "ride-n1, commute-n1")
  (syn-struct((root($var0))(cat(n))
  (pp-adjunct((opt(+))(root(by))(cat(pre)))
  (obj((root($var1))(cat(np))))))
  (pp-adjunct((opt(+))(root(to))(cat(pre)))
  (obj((root($var2))(cat(np))))))
  (sem-struct(drive(agent(value(^$var1)))
  (instrument(value(^$var2))))))
)
```

Adjectival senses

The sem-struct of the adjectival sense will be generated from the INSTRUMENT case role fillers

of “drive-v1” by specifying the variable for the INSTRUMENT property inverse and assigning a high probabilistic (EPISTEMIC) modality value to the event DRIVE. Because CAR, BUS, TRACTOR and TRUCK are already specified as INSTRUMENT fillers of DRIVE in the ontology, they will be omitted from the variable restriction in the adjective sem-struct for reasons of redundancy.

Name assignment and synonym check procedures have limited application for adjective derivation for reasons described in 2.2(2); so the name “drivable”, is assigned manually. The complete sense is listed below:

```
(drivable-adj
  (cat(adj))
  (anno(def "capable of being driven: of
  cars")(comments "")(ex "Bill sold a completely
  drivable car. This old car is still drivable"))
  (synonyms "")
  (syn-struct((root($var0))(cat(adj))
  (syn-np((root($var1))(cat(np))))))
  (syn-struct1((syn-np((root($var1))(cat(np))))
  (vp((root(be))(cat(v))))(adj-
  comp((root($var0))(cat(adj))))))
  (sem-struct(^$var1(restricted-to(car bus
  tractor truck))(instrument-of(default(drive(epistemic
  (value(0.8))))))))
)
```

4 References

- [1] C. Hempelmann, J. Taylor; and V. Raskin, “Application-guided ontology engineering” In Proc. ICAI’10, Las Vegas, July 2010.
- [2] J. Katz, and J. Fodor. “The Structure of a Semantic Theory”. In *Language*, vol. 39, No. 2, pp. 170-210, April-June 1963
- [3] S. Nirenburg and V. Raskin, *Ontological Semantics*. Cambridge, MA: MIT Press, 2004.
- [4] V. Ooi. *Computer Corpus Lexicography*. Edunburg University Press, 1998.
- [5] J. Preiss, T. Briscoe, and A. Korhonen. “A System for large scale acquisition of verbal, nominal, and adjectival subcategorization frames from corpora”. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. Prague, Czech Republic, 2007.
- [6] M. Petrenko, “Lexicon Management in Ontological Semantics”. In *Papers from the Annual international Conference “Dialogue 2010, Moscow, Russia 2010*, in press.
- [7] V. Raskin, J. M. Taylor, and C. F. Hempelman, “Guessing and knowing: Two approaches to semantics in natural language processing,”. In *Papers from the Annu. Int. Conf. “Dialogue”*, Moscow, 2010, vol. 9, no. 16, pp. 642-650.
- [8] J. Taylor, C.F. Hempelmann, and V. Raskin, “On an automatic acquisition toolbox for ontologies and lexicons” In Proc. ICAI’10, Las Vegas, USA, July 2010.