Abstract

In this paper, we propose a fine grained classification of English adjectives geared at modeling the distinct inference patterns licensed by each adjective class. We show how it can be implemented in description logic and illustrate the predictions made by a series of examples. The proposal has been implemented using Description logic as a semantic representation language and the prediction verified using the DL theorem prover RACER.


topics: Textual Entailment, Adjectival Semantics

1 Introduction

Understanding a text is one of the ultimate goals of computational linguistics. To achieve this goal, systems need to be developed which can construct a meaning representation for any given text and which furthermore, can reason about the meaning of a text. As is convincingly argued in (rte, 2005), one of the major inference task involved in that reasoning is the entailment recognition task:

Does text $T_1$ entail text $T_2$?

Indeed entailment recognition can be used to determine whether a text fragment answers a question (e.g., in question answering application), whether a query is entailed by a relevant document (in information retrieval), whether a text fragment entails a specific information nugget (in information extraction), etc.

Because the Pascal RTE challenge focuses on real text, the participating systems must be robust that is, they must be able to handle unconstrained input. Most systems therefore are based on statistical methods (e.g., stochastic parsing and lexical distance or word overlap for semantic similarity) and few provide for a principled integration of lexical and compositional semantics. On the other hand, one of the participant teams has shown that roughly 50% of the RTE cases could be handled correctly by a system that would adequately cover semantic entailments that are either syntax based (e.g., active/passive) or lexical semantics based (e.g., bicycle/bike). Given that the overall system accuracies hovered between 50 and 60 percent with a baseline of 50%\(^1\), this suggests that a better integration of syntax, compositional and lexical semantics might improve entailment recognition accuracy.

In this paper, we focus on the case of adjectives and explore the entailment patterns that are supported by the interaction of their lexical and of their compositional semantics. We start by defining a classification schema for adjectives based on their syntactic and semantic properties. We then associate with each class a set of axioms schemas and of semantic construction rules and we show that these correctly predicts the observed entailment patterns. For instance, the approach will account for the following (non)-entailment cases:

1. a. John frightened the child
   $\models$ The child is afraid

   b. John is an alleged murderer
   $\models$ Peter claims that John is a murderer
   $\not\models$ John is a murderer

   c. This is a fake bicycle
   $\models$ This is a false bike
   $\models$ This is not a real bike
   $\not\models$ This is a bike

\(^1\)50% of the cases were true entailment and 50% were false ones, hence tossing a coin would get half of the cases right.
The approach is implemented using Description Logic as a semantic representation language and tested on a hand-built semantic test suite of approximately 1,000 items. In the latter part of the paper we discuss this testsuite and the philosophy behind it.

2 A fine grained classification for adjectives

As mentioned above, we semantically classify adjectives based on their lexical, their model theoretic and their morpho-derivational properties. To facilitate the link with compositional semantics (the construction of a meaning representation for sentences containing adjectives), we also take into account syntactic properties such as the predicative/attributive or the static/dynamic distinction. We now detail each of these properties. The overall categorisation system is given in Figure 1.

2.1 Model theoretic properties

The main criteria for classification are given by (Kamp, 1975; Kamp and Partee, 1995) semantic classification of adjectives which is based on whether it is possible to infer from the Adj+N combination the Adj or the N denotation.

**Intersective adjectives** (e.g., red) licence the following inference patterns:

\[ A + N \models A \]
\[ A + N \models N \]

For instance, if *X is a red car* then *X is a car* and *X is red*

**Subsective adjectives** (e.g., big) licence the following inference pattern:

\[ A + N \models N \]

For instance, if *X is a big mouse*, then *X is a mouse* but it is not necessarily true *X is big*

**Privative adjectives** licence the inference pattern:

\[ A + N \models \neg N \]

For instance, if *X is a fake gun* then *X is not a gun*

For instance, if *X is an alleged murderer* then it is unknown whether *X is a murderer or not*

2.2 Lexical semantics

From the lexical semantics literature, we take one additional classification criterion namely antonymy. This term covers different kinds of opposite polarity relations between adjectives namely, binary opposition, contraries and multiple oppositions.

**Binary oppositions** covers pairs such as *wet/dry* which license the following inference pattern:

\[ A_1 \equiv \neg A_2 \land \neg A_1 \equiv A_2 \]

So that in particular:

\[ wet \equiv \neg dry \land \neg wet \equiv dry \]

**Contraries** are pairs such as *long/short* where the implication is unidirectional:

\[ A_1 \models \neg A_2 \land \neg A_1 \not\models A_2 \]
\[ A_2 \models \neg A_1 \land \neg A_2 \not\models A_1 \]

and in particular:

\[ long \models \neg short \land \neg long \not\models short \]
\[ short \models \neg long \land \neg short \not\models long \]

**Multiple oppositions** involve a finite set of adjectives (e.g., linguistic/economic/mathematical/) which are pairwise mutually exclusive. For a set of opposed adjectives \( A_1 \ldots A_n \), the following axioms schemas will be licensed:

\[ \forall i, j \text{ s.t. } 1 \leq i, j \leq n \text{ and } i \neq j \]

\[ A_i \models \neg A_j \text{ and } \neg A_i \not\models A_j \]

2.2.1 Derivational morphology

We also take into account related forms that is, whether there exists a noun or a verb that is semantically related to the adjectives being considered. Moreover, for nominalizations we distinguish whether the morphologically related noun is an event noun or a noun denoting a theta role of the related verb.

As we shall see, this permits capturing entailment relations between sentences containing morphodervational variants such as for instance:
(2) a. John is asleep (Adjective → Verb)  
   \[ \models \text{John sleeps} \]

b. John is absent (Adj. → θ-role Noun)  
   \[ \models \text{John is the absentee} \]

c. John is deeply asleep (Adj. → evt N.)  
   \[ \models \text{John’s sleep is deep} \]

2.2.2 Syntactic properties

To better support the syntax/semantic interface, we refine the semantic classes distinguishable on the basis of the above criteria with the following syntactic ones taken from (Quirk et al., 1985).

Attributiveness/Predicativeness. English adjectives can be divided in adjectives which can be used only predicatively (such as alone), adjectives which can be used only attributively (such as mechanical in mechanical engineer) and adjectives which can be used in both constructions such as red.

Modifiability by very. We distinguish between adjectives such as nice which can be modified by very (i.e. very nice) and adjectives such as alleged which cannot (*very alleged).

Staticity/Dynamicity. Dynamic adjectives can be used in imperative constructions and in the progressive form (Be reasonable, He is being reasonable), static adjectives cannot (*Be short, He is being short).

3 Semantic Classes and textual entailment recognition

As Figure 1 shows, the proposed classification includes 15 adjective classes, each with distinct syntactic and semantic properties.

To account for these differences, we define for each class a set of axiom schemas capturing the model theoretic, lexical semantics and morpho-derivational properties of that class. Based on some basic syntactic patterns, we then show that these axioms predict the observed textual entailment patterns for that class.

Before we illustrate this approach by means of an example, we first show how we capture logical entailment between NL semantic representations in a description logic setting.

3.1 Using description logic to check entailment between NL sentences

As argued in (Gardent and Jacquey, 2003), description logic (DL) is an intuitive framework within which to perform lexical reasoning: it is efficient (basic versions of description logics are decidable), it is tailored to reason about complex taxonomies (taxonomies of descriptions) and it is equipped with powerful, freely available automated provers (such as RACER, (Volker Haarslev, 2001)). For these reasons, we are here exploring a DL encoding of the entailment recognition task for the set of examples we are considering. The particular language we assume has the following syntax.

\[ C, D \rightarrow A \mid \top \mid \bot \mid \neg A \mid C \cap D \mid C \cup D \mid \forall R.C \mid \exists R.C \]

The semantics of this language is given below with \( \Delta \) the domain of interpretation and \( I \) the interpretation function which assigns to every atomic concept \( A \), a set \( A_I \subseteq \Delta \) and to every atomic role \( R \) a binary relation \( R_I \subseteq \Delta \times \Delta \).

\[
\begin{align*}
\top_I &= \Delta \\
\bot_I &= \emptyset \\
(\neg A)_I &= \Delta \setminus A_I \\
(C \cap D)_I &= C_I \cap D_I \\
(C \cup D)_I &= C_I \cup D_I \\
(\forall R.C)_I &= \{ a \in \Delta \mid \forall b \in R_I \rightarrow b \in C_I \} \\
(\exists R.C)_I &= \{ a \in \Delta \mid \exists b \in C_I \land (a, b) \in R_I \}
\end{align*}
\]

Now one basic problem with using DL to check entailment between NL expressions, is that DL formulae are “directional” in that they refer to a given set of individuals. For instance the sentence The boat is floating might be represented by either of the two formulae given in 3 but these two formulae do not stand in an entailment relation (since they refer to different kind of objects namely floating event of a boat in 3a and boats that float in 3b).

(3) a. \( \text{float} \sqsubseteq \exists \text{theme.boat} \)

b. \( \text{boat} \sqsubseteq \exists \text{theme}^{-1}.\text{float} \)

To remedy this shortcoming, we introduce the notion of a rotation. Given a DL formula which only contains conjunction (disjunction is translated in DL as different formulas)

\[ \Phi = \bigcap_{i=1,n} \text{Event}_i \cap \bigcap_{j=1,m} \exists R_j, \text{Type}_j \]

a rotation of this formula is defined as:

1. \( \Phi \)
2. \( \forall j \in \{1, \ldots, m\} : \text{Type}_{j} \cup \exists R_{j}^{-1}.(\bigcap_{i=1,n} \text{Event}_i \cap \bigcap_{k<i,j<k<m} \exists R_k, \text{Type}_k) \)
so that the formula:
Event \sqcap \text{Event}_2 \sqcap \ldots \sqcap \text{Event}_n \sqcap \text{R}_1.\text{Type}_1 \sqcap \text{R}_2.\text{Type}_2 \ldots \sqcap \text{R}_n.\text{Type}_n

\subseteq \text{Event}_1

1. \text{Type}_1 \sqcap \text{R}_1^{-1}.(\text{Event} \sqcap \text{R}_2.\text{Type}_2 \ldots \sqcap \text{R}_n.\text{Type}_n) \subseteq \text{Type}_1

2. \text{Type}_2 \sqcap \text{R}_2^{-1}.(\text{Event} \sqcap \text{R}_1.\text{Type}_1 \ldots \sqcap \text{R}_n.\text{Type}_n) \subseteq \text{Type}_2

\ldots

n. \text{Type}_n \sqcap \text{R}_n^{-1}.(\text{Event} \sqcap \text{R}_1.\text{Type}_1 \ldots \sqcap \text{R}_{n-1}.\text{Type}_{n-1}) \subseteq \text{Type}_n

So for example, the sentence Mary knows that John is the inventor of the radio will be represented as a predicate logic formula

\exists x_1.\text{mary}(x_1) \land \exists x_2.\text{john}(x_2) \land \exists x_3.\text{radio}(x_3) \land \exists e_1.\text{know}(e_1) \land \exists e_2.\text{agent}(e_2) \land \exists e_3.\text{topic}(e_3) \land \exists e_4.\text{invent}(e_2) \land \exists e_5.\text{agent}(e_5) \land \exists e_6.\text{patient}(e_6)

the denotation of this PL formula corresponds to the set of individual \( \{x_1, x_2, x_3\} \cup \{e_1, e_2\} \).

The corresponding DL representation will be the underspecified representation

\text{know} \sqsubseteq \text{agent}.\text{mary} \sqsubseteq \text{topic}.(\text{invent} \sqsubseteq \text{agent}.\text{john} \sqsubseteq \text{patient}.\text{radio})

the denotation of which corresponds to the set \( \{e_1\} \) and all its rotations which permitt to access the other sets of individuals asserted in the sentence:

Rotation_0: know \sqsubseteq \text{agent}.\text{mary} \sqsubseteq \text{topic}.(\text{invent} \sqsubseteq \text{agent}.\text{john} \sqsubseteq \text{patient}.\text{radio})

Rotation_1: \text{mary} \sqsubseteq \text{agent}^{-1}.(\text{know} \sqsubseteq \text{topic}.(\text{invent} \sqsubseteq \text{agent}.\text{john} \sqsubseteq \text{patient}.\text{radio})

Rotation_2: (\text{invent} \sqsubseteq \text{agent}.\text{john} \sqsubseteq \text{patient}.\text{radio}) \sqsubseteq \text{topic}^{-1}.(\text{know} \sqsubseteq \text{agent}.\text{mary})

Finally, we say that an arbitrary formula/representation \( \Phi_1 \) implies the formula \( \Phi_2 \) iff it is possible to find a rotation \( Rotation_i \) of \( \Phi_1 \) the denotation of which describes a subset of the denotation of \( \Phi_2 \):

Definition

\( \Phi_1 \models \Phi_2 \) iff \( \exists i.\ Rotation_i(\Phi_1) \subseteq \Phi_2 \)  \hspace{1cm} (1)

3.2 Example class axioms and derivations

We now illustrate our approach by looking at two classes in more detail namely, class 1 and class 8.

3.2.1 Class 1

Syntactically, Class 1 contains adjectives like adrift, afloat, aground which can only be used predicatively, are non gradable and cannot be modified by very. Semantically, they are intersective adjectives which enter in multiple opposition relations with other adjectives. They are furthermore morphologically derived from verbs and can be nominalized. To reflect these semantic properties we use the following axioms.

Model theoretic semantics. Adjectives of class 1 are intersective adjective. They will thus licence the corresponding inference patterns namely:

\[
A + N \models A \hspace{1cm} (2)
\]

\[
A + N \models N \hspace{1cm} (3)
\]

Lexical semantics. Adjectives of class 1 enter in multiple opposition relations. Hence For
instance:

afloat \models \neg aground \land \neg aground \not\models aground
aground \models \neg afloat \land \neg aground \not\models afloat
sunken \models \neg afloat \land \neg afloat \not\models sunken
afloat \models \neg sunken \land \neg sunken \not\models afloat

Morpho-derivational semantics. Adjectives in Class 1 can be related to both nouns and verbs. This is encoded in the following axiom schemas:

MDR 1. Adj1 \sqsubseteq \neg Adj2
    e.g., afloat \sqsubseteq \neg sunken
MDR 2. Adj1 \equiv \exists Theme^{-1}.V1
    If Adj1 is related to V1
    e.g., afloat \equiv \exists Theme^{-1}.float
MDR 3. V1 \sqsubseteq \neg V2
    If V1 = Anto(V2)
    e.g., float \sqsubseteq \neg sinking
MDR 4. N1 \equiv V1
    If Adj1 is related to an event denoting N1
    e.g., floating \equiv float
MDR 5. N1 \sqsubseteq \neg N2
    If N1 is an antonym of N2
    e.g., floating \sqsubseteq \neg sinking
MDR 6. N11 \equiv \exists Theme^{-1}.V1
    If Adj1 is related to a noun N11 denoting the theme role of the verb V1
    e.g., floater \equiv \exists Theme^{-1}.float

We make the following assumptions about the syntax/semantic interface that is, about the semantic representations associated with given sentence patterns.

SCR 1. NP toBe Adj

SCR 2. NP toBe clearly Adj

SCR 3. Ni[+event] of NP is clear

SCR 4. Ni[-event] is clear

SCR 5. NP toBe V[+ing].

Given the above axiom schemas and semantic constructions rules, the following inference patterns can be handled:

1. ADJ1 + N \models N
   Ex. This boat is afloat. \models This is a boat.
2. ADJ1 + N \models ADJ1
   Ex. This boat is afloat. \models This is afloat.
3. ADJ1 + N \not\models \neg N
   Ex. The boat is afloat. \not\models This not a boat.
4. ADJ1 + N \models \neg ADJ2 \cap N
   Ex. The boat is afloat. \models The boat is not sunken.
5. \neg ADJ1 + N \not\models ADJ2 \cap N
   Ex. The boat is not afloat. \not\models The boat is sunken.
6. ADJ1 + N \models N \cap \exists Theme^{-1}.V1
   Ex. The boat is afloat. \models The boat is the floater.
7. ADJ1 + N \models V1 \cap \exists Theme.N
   Ex. This boat is clearly afloat. \models The floating of the boat is clear.
8. ADJ1 + N \models N \cap \exists Theme^{-1}.N
   Ex. This boat is clearly afloat. \models The floating of the boat is clear (or the boat is the floating object).
9. \neg (ADJ1 + N) \models \neg (V1 \cap \exists Theme.N) \not\models \neg N
   Ex. This is not a floating boat. \not\models This is not a boat.
10. \neg (ADJ1 + N) \not\models \neg ADJ1
    Ex. This is not a floating boat. \not\models This is not afloat.
11. \neg (ADJ1 + N) \not\models \neg V1
    Ex. This is not a floating boat. \not\models This is not floating.
12. \neg (ADJ1 + N) \not\models \neg N
    Ex. This is not a floating boat. \not\models This is not a boat.
13. \neg (ADJ1 + N) \not\models \neg ADJ1
    Ex. This is not a floating boat. \not\models This is not afloat.
14. \neg (ADJ1 + N) \not\models \neg \exists Theme^{-1}.V1
    Ex. This is not a floating boat. \not\models This is not the floater.
15. \neg (ADJ1 + N) \not\models \neg \exists Theme.N
    Ex. This is not a floating boat. \not\models This is not a floating.

In the inference patterns 10 to 15, the negation of the adjective-noun compound \neg (ADJ1 + N) is syntactically blocked, as the adjectives in this class are used predicative only, however the equivalent representation V1 \cap \exists Theme.N can be used to motivate the inferences.

The following show in more detail how the first three of the above (non) entailments are recognised.
Example 1.

(4) a. The boat is afloat.

b. |= The boat is floating.

<table>
<thead>
<tr>
<th></th>
<th>4a</th>
<th>\equiv \text{Boat} \sqcap \text{AfBat} \quad \text{(by SCR 1)}</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4b</td>
<td>\equiv \text{Float} \sqcap \exists \text{Theme}.\text{Boat} \quad \text{(by SCR 5)}</td>
</tr>
<tr>
<td>B</td>
<td>AfBat</td>
<td>\equiv \exists \text{Theme}^{-1}.\text{Float} \quad \text{(by MDR 2)}</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>\equiv \text{Boat} \sqcap \exists \text{Theme}^{-1}.\text{Float} \quad \text{(from A and C)}</td>
</tr>
<tr>
<td>D</td>
<td>E</td>
<td>\equiv \text{Boat} \quad \text{(By Defn 1)}</td>
</tr>
<tr>
<td>D</td>
<td>J</td>
<td>\equiv B \quad \text{(By Defn 1)}</td>
</tr>
</tbody>
</table>

Example 2.

(5) a. The boat is afloat.

b. |= The boat is the floater.

| | 5a | \equiv \text{Boat} \sqcap \text{AfBat} \quad \text{(by SCR 1)} |
| A | 5b | \equiv \text{Boat} \sqcap \exists \text{Theme}^{-1}.\text{float} \quad \text{(by SCR 4)} |
| B | AfBat | \equiv \exists \text{Theme}^{-1}.\text{Float} \quad \text{(by MDR 2)} |
| C | A | \equiv B \quad \text{(from B and C)} |
| D | E | \equiv B \quad \text{(By Defn 1)} |

Example 3.

(6) a. The boat is afloat.

b. |= The boat is not sinking.

| | 6a | \equiv \text{Boat} \sqcap \text{AfBat} \quad \text{(by SCR 1)} |
| A | 6b | \equiv \sim \text{sink} \sqcap \exists \text{Theme}.\text{boat} \quad \text{(by SCR 5)} |
| B | AfBat | \equiv \exists \text{Theme}^{-1}.\text{Float} \quad \text{(by MDR 2)} |
| C | E | \equiv \text{Boat} \sqcap \exists \text{Theme}.\text{boat} \quad \text{(By Defn 1)} |
| D | E | \equiv \text{Boat} \quad \text{(by MDR 1)} |

3.2.2 Class 8.

Class 8 contains adjectives like big, fast, tall, deep which can be used attributively and predicatively, are gradable, can be modified by very. Semantically, they are classified as subsective adjectives and their antonyms are contraries. They are morphologically related to nouns which describe the particular property denoted by the adjectives and to nouns of which they are attributes.

Model theoretic semantics. Adjectives of class 8 are subsective adjective. They will thus license the corresponding inference patterns namely:

\[ A + N \not\models A \quad \text{(4)} \]
\[ A + N \models N \quad \text{(5)} \]

Lexical semantics. The Adjectives of class 8 enter in contrary opposition relations. Hence, the following axioms schemas will be licensed:

\[ A_i \models \neg \text{Anto}(A_i) \text{ and } \neg A_i \not\models \text{Anto}(A_i) \quad \text{(6)} \]

For instance:

\[ \text{long} \models \neg \text{small} \land \sim \text{long} \not\models \text{small} \]
\[ \text{deep} \models \neg \text{shallow} \land \sim \text{deep} \not\models \text{shallow} \]

Morpho-derivational semantics. Adjectives in Class 8 can be related to nouns but not to verbs. This is encoded in the following axiom schemas:

MDR 1. Adj1 \sqsubseteq Adj2 \quad \text{If Adj1 = Anto(Adj2)}
Ex. tall \sqsubseteq short

MDR 2. Adj1 \sqsubseteq \exists \text{has\_property}.(N1 \sqsubseteq \exists \text{has\_measure}.Top)
If Adj1 is related to a noun N1 denoting the property described by Adj1
Ex. tall \sqsubseteq \exists \text{has\_property}.(tallness \sqsubseteq \exists \text{has\_measure}.Top)

MDR 3. N1 \sqsubseteq N2 \quad \text{If N1 = Anto(N2)}
Ex. tallness \sqsubseteq \neg \text{shortness}

MDR 4. N1 \sqsubseteq N' \sqsubseteq \exists \text{has\_value}.Adj1 \sqsubseteq \exists \text{has\_measure}.Top
If Adj1 is an attribute of the noun N' denoting the property described by Adj1
Ex. tallness \sqsubseteq \exists \text{has\_value}.tallness \sqsubseteq \exists \text{has\_measure}.Top

MDR 5. N2 \sqsubseteq N' \sqsubseteq \exists \text{has\_value}.Adj2 \sqsubseteq \exists \text{has\_measure}.Top
If Adj2 is an attribute of the noun N' denoting the property described by Adj2
Ex. shortness \sqsubseteq \exists \text{has\_value}.shortness \sqsubseteq \exists \text{has\_measure}.Top

MDR 6. N1 \sqsubseteq N' \quad \text{If N1 is a hyponym of N'}
Ex. tallness \sqsubseteq \neg \text{height}

MDR 7. N2 \sqsubseteq N' \quad \text{If N2 is a hyponym of N'}
Ex. shortness \sqsubseteq \neg \text{height}

MDR 8. Adj11 \sqsubseteq Adj1 \quad \text{If Adj1 is a scalar attribute with value less then Adj11 (hyponymy is not defined for adjectives)}
Ex. giant \sqsubseteq \neg \text{tall}

We make the following assumptions about the semantic representations associated with basic sentence patterns.

SCR 1. NP toBe Adj
NP \sqcap \exists \text{has\_property}.(N1 \sqsubseteq \exists \text{has\_measure}.NP)

SCR 2. That toBe Det Adj NP
NP \sqcap \exists \text{has\_property}.(N1 \sqsubseteq \exists \text{has\_measure}.NP)

SCR 3. NP toBe clearly Adj
NP \sqcap \exists \text{has\_property}.(N1 \sqsubseteq \exists \text{has\_measure}.NP)

SCR 4. N1 of NP is clear
NP \sqcap \exists \text{has\_property}.(N1 \sqsubseteq \exists \text{has\_measure}.NP)

SCR 5. The Adj N' of NP
NP \sqcap \exists \text{has\_property}.(N' \sqsubseteq \exists \text{has\_value}.Adj \sqsubseteq \exists \text{has\_measure}.NP)
Given the above axioms, the following inference patterns can be handled:

1. ADJ1 + N \models N
   This animal is tall. \models This is an animal.

2. ADJ1 + N \not\models ADJ1
   This animal is tall. \not\models This is tall.

3. ADJ1 + N \not\models \neg N
   This animal is tall. \not\models This not an animal.

4. ADJ1 + N \models \neg ADJ2 \sqcap N
   This animal is tall. \models This animal is not small.

5. \neg ADJ1 + N \not\models ADJ2 \sqcap N
   This animal is not tall. \not\models This animal is small.

6. ADJ1 + N \models N \sqcap \exists \text{has\_property.}(N \sqcap \exists \text{has\_measure}.N)
   This animal is tall. \models This animal has tallness.

7. ADJ1 + N \not\models \exists \text{has\_property.}(N \sqcap \exists \text{has\_measure}.\text{Top})
   This animal is tall. \models tallness.

8. ADJ1 + N \models N \sqcap \exists \text{has\_property.}(N' \sqcap \exists \text{has\_value}.\text{ADJ1} \sqcap \exists \text{has\_measure}.N)
   This animal is tall. \models This animal has a tall stature.

9. ADJ1 + N \not\models \exists \text{has\_property.}(N' \sqcap \exists \text{has\_value}.\text{ADJ1} \sqcap \exists \text{has\_measure}.\text{Top})
   This animal is tall. \not\models tall stature.

10. \neg (ADJ1 + N) \not\models \neg N
    This is not a tall animal. \not\models This is not an animal.

11. \neg (ADJ1 + N) \not\models \neg ADJ1
    This is not a tall animal. \not\models This is not tall.

12. \neg (ADJ1 + N) \not\models \exists \text{has\_property.}(N \sqcap \exists \text{has\_measure}.N)
    This is not a tall animal. \not\models This animal has not tallness.

13. \neg (ADJ1 + N) \not\models \exists \text{has\_property.}(N' \sqcap \exists \text{has\_value}.\text{ADJ1} \sqcap \exists \text{has\_measure}.N)
    This is not a tall animal. \not\models This animal has not a tall stature.

Example 1.

(7) (a) John is a 2 meter tall man. \models (b) John is 2 meter tall.

(8) (a) John is a 2 meter tall man. \not\models (b) John is a tall man.

4 Implementation

For each of the 15 classes, we have specified a set of axioms schemas, some basic semantic construction rules and a set of inference patterns which could be deduced to follow from both of these. The axioms schemas were implemented in Description Logic using RACER and for each inference pattern identified, the corresponding Description Logic query was checked to verify that the proposed axioms and semantic construction rules did indeed correctly predict the deduced inference patterns.

5 Further work and evaluation

The main contribution of this work is a detailed analysis of the interactions between derivational morphology, lexical and compositional semantics and of their impact on the entailment patterns licensed by sentences containing adjective or their related nouns/verbs.

To turn this analysis into a computational system, its components need to be integrated into a
The boat is afloat.
The boat is floating.

This is not a rectangular table.
This is a round table.

The line is 2 meter long.
The length of the line is 2 meter.

The treasurer is present.
This is the present treasurer.

The idea behind this test suite is similar to that underlying the creation of the TSNLP (Test suite for natural language processing) or the Eurotra testsuites namely, to provide a benchmark against which to evaluate and compare existing semantic analyzers. Thus this test suite illustrates the semantic and syntactic behaviour of adjectives and their related verbs/nouns with respect to textual entailment. One could imagine other test suites illustrating the semantic behaviour of verbs, of quantifiers, of discourse connectives, etc. Just as the TSNLP still proves useful in supporting the development of new symbolic parsers/grammars, hand built test suites of artificial examples might prove useful in improving the accuracy of semantic analyser wrt textual entailment. Indeed the Pascal RTE challenge has shown that existing systems fares rather poorly at the textual entailment task. Providing a set of hand crafted semantic test suites might help in remedying this shortcoming.

Beside implementing and evaluating the analysis of adjectives presented in this paper, we are also working on refining this analysis by combining it with a detailed analysis of noun semantics so as to handle (non) entailments such as:

Lyon is the gastronomical capital of France
≠ Lyon is the capital of France

References


