Semantic Representations of English Adjectives-Noun Combinations for NLP

MieAe Jung
(Indiana University)

Jung, MieAe. 2002. Semantic Representations of English Adjective-noun Combinations for NLP. *The Linguistic Association of Korea Journal, 10*(4), 93-116. This study deals with the semantic representations of English adjective-noun combinations for Natural Language Processing. For this purpose, the program was implemented in Prolog+CG. The basic idea of this study is that in order to account for the meaning of adjective-noun combinations, the type definition for an adjective should be declared. This paper shows that the meaning of adjective-noun combinations can be revealed in three ways: maximalJoin between the sense graph(s) for the adjective and the concept of the noun, maximalJoin between the sense graph for the adjective and the sense graph for the noun, and maximalJoin between the sense graph for the adjective and the schema for the noun in adjective-noun constructions.

Key words: adjective-noun combinations, NLP, Prolog+CG, type definition, schema.

1. Introduction

Representing the meaning of adjectives is one of the challenging issues in natural language processing. One of the well-known reasons is 'plasticity' of adjectival meaning or 'non-compositionality'. In other words, the meaning of an adjective shifts with the meaning of the noun it modifies, depending on what property of that noun the adjective pertains to (Raskin and Nirenburg, 1995).

In many cases, the analysis of adjective-noun combinations is somewhat complicated. It thus appears that semantic analyses and lexico-semantic descriptions of adjectives have been rare in natural language processing. But as Raskin and Nirenburg (1995) mention, this situation is changing rapidly. As computational semantics operates on larger-scale systems, a larger lexicon is needed for the lexical entries with all their possible lexical categories in them. For this purpose, computational semanticists and lexicographers are paying more attention to the adjectival category, a previously neglected one.

Motivated by this trend, this study focuses on the semantic descriptions of adjectives and suggests an expressive method of representing the meaning of adjective-noun combinations.

In what follows, we will briefly review some of the previous approaches of computational semantics to adjectives.

2. Previous Work

Adjectives can be classified in many ways from different perspectives: syntactic, semantic, and ontological. Syntactically, adjectives can be classified with respect to their function, complementation, and alternation. Semantically, according to their logical behavior, adjectives can also be classified with respect to other semantic features such as aspect or gradation (Bouillon and Viegas, 1999).

In this section, we will review an ontological approach and a semantic network approach to adjectives.

In an ontological approach like MikroKosmos, each lexical entity shows the lexical mapping from language unit to ontological concepts in the LEX-MAP part of the SEM-STRUC zone and the linking between the syntactic and the semantic structure in the SYN-STRUC zone. Raskin and Nirenburg (1995) propose a microtheory for adjectival meaning which should be included in a computational lexicon. Their goal is to develop a method for describing the semantics of adjectives in a text. Their work on adjectives forms a microtheory used by the MikroKosmos semantic analyzer, and assumes that the lexicon is the locus of the microtheory of adjectival meaning in MikroKosmos.

In this approach, English adjectives are classified into four subclasses:

attitude-based (*good, superb, important*), numerical scale (*big, heavy, forte, pricey, opulent, ripe, young*), literal scale (*red, magenta, oval, front, backward*), and member (*authentic, fake, nominal*). Raskin and Nirenburg (1995) argue that many adjectives do not modify semantically the nouns that they modify syntactically, the syntactic behavior of an adjective does not determine its lexical meaning, and thus the attribute/predicative distinction has no semantic significance. The following displays the lexical entry for *good*.

```
(1)      (good
              (good-Adj1
                (CAT adj)
                (SYN-STRUC
                   (1 ((root $var1)
                          (cat n)
                          (mods ((root $var0)))))
                   (2 ((root $var0)
                          (cat adj)
                          (subj ((root $var1)
                                  (cat n))))))
                (SEM-STRUC
                        (LEX-MAP
                          (attitude
                               (type evaluative)
                               (attitude-value (value (> 0.75))
                                   (relaxable-to (value (> 0.6))))
                               (scope ^$var1)
                             (attributed-to *speaker*))))))
```

In their microtheory, *good* selects a property of a noun and assigns its high value on the evaluation scale associated with the property of the noun.

WordNet, a semantic network approach, represents the largest publicly available online lexical resources: English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets. Modifiers in WordNet are coded by means of various structural features and relational pointers. These features and pointers are interpreted by the interface, which represents the information to the user in a direct way. The system divides adjectives into two major classes: descriptive (such as *big, interesting,* and *possible*) and relational (such as *presidential* and *nuclear*). WordNet contains pointers between descriptive adjectives expressing a value of an attribute and the noun by which that attribute is lexicalized. The interaction between adjectives and noun are not prestored but computed as needed by an on-line interpretation process (Fellbaum *et al.*, 1993; Miller, 1998).

In WordNet, each sense in an entry is determined by a 'synset', a set of synonyms. For instance, 25 synsets are given for adjective *good* by WordNet 1.7.1 (http://www.cogsci.princeton.edu/cgi-bin/webwn1.7.1). The following show three of the synsets:

(2) a. good (vs. bad) -- (having desirable or positive qualities especially those suitable for a thing specified; "good news from the hospital"; "a good report card"; "when she was good she was very very good"; "a good knife is one good for cutting"; "this stump will make a good picnic table"; "a good check"; "a good joke"; "a good exterior paint"; "a good secretary"; "a good dress for the office")
b. full, good -- (having the normally expected amount; "gives full measure"; "gives good measure"; "a good mile from here")
c. good (vs. evil) -- (morally admirable)

Due to its vast coverage of lexico-semantic information as shown in (2) above, WordNet can be a useful resource to develop a word sense disambiguation algorithm.

The present study will take advantage of the characteristics of the two approaches mentioned above: the ontological information and the synonym sets of adjectives.

3. Semantic Representations for Adjective-Noun Combinations in Prolog+CG

This paper deals with the semantic representations of adjective-noun combinations for natural language processing. To this end, programs are implemented in Prolog+CG 2.5, an object-oriented logic programming language based on Conceptual Graphs (CGs).

CGs (Sowa, 1984, 1991, 1992, 2000) have been used in many fields of artificial intelligence (AI), especially natural language processing and knowledge-based systems. CGs provide the expressive power of an advanced knowledge representation language, *i.e.*, advanced semantic nets, type hierarchies, schemata, the notion of context, and the like.

Prolog+CG (Kabbaj *et al.*, 2001) is a contextual extension of Prolog in the sense that it integrates notions like objects and inheritance. The integration of Prolog, object-oriented programming, CG, and Java provides a powerful developmental environment for the creation of knowledge-based applications. However, although Prolog+CG provides rich manipulating operations of CG with the expressive power, and there has been research to resolve linguistic phenomena such as a narrative analysis (Schärfe, 2002) and an indirect anaphora resolution (Jung, 2002), there has been no practical research to represent the meaning of adjective-noun combinations in the language. Thus, this paper will show how to represent the meaning of these combinations by implementing Prolog+CG.

CGs include relations, structures, and guidelines for representing sentences in natural language. Their basic principle is that content words map to concept nodes, and function words like prepositions and conjunctions map to relation nodes. Ordinary nouns, verbs, adjectives, and adverbs map to type labels in a concept node: lady⇒[Lady], dance ⇒[Dance], happy⇒[Happy] (Sowa, 1991). In this mapping principle, adjectives can exist independently without any other connection to another concept. But in Sowa (1999), the notation to describe adjectives slightly changed. Nouns become type labels, while adjectives become monadic conceptual relations. Thus, for instance, the adjective *available* can be represented by [Available] linked by the attribute relation (Attr), i.e., [⊤]->(Attr)-> [Available], and since [⊤] is the universal type, it may be any concept of any type.

However, if [Available] can be linked to any concept of any type, the formula does not guarantee that the graph is canonical. For instance, if we follow the notation above, the formula for *green* will look like [⊤]->(Colr)->[Green]. However, since the conceptual type of [⊤] is unspecified, the formula cannot block anomalous combinations such as [Idea]->(Colr)->[Green].

For this reason, the present study suggests that formulae for adjectives should be canonical, where concepts that can be linked to the concept for an adjective are specified. For instance, since the adjective *green* will be represented by the graph [Object]->(Colr)->[Green] and [Idea] is not [Object], *green* cannot be joined with *idea*.

The representations suggested in this study satisfy the compositionality requirement in that the meaning of an adjective-noun combination is obtained through the composition of the meaning of the adjective and that of the noun. A graph resulting from that composition is well-formed, since the sense graph for an adjective must be canonical and a canonical graph rules out anomalies by enforcing selectional constraints.

Now, let us take a look at the example *red car*. (3) and (4) below show two components of Prolog+CG: the program pane and the interface pane.

A program in Prolog+CG is composed of type hierarchies which present ontological information of conceptual types and conceptual graphs as seen in (3) below. (4) shows the interface between the user and the system, where if the user asks the system a question, the system provides the answers to the request.

(3)
```
Universal > Object, Person, Property.
Object > Vehicle.
Vehicle > Car.
Property > Visual_Property, Tactile_Property.
Visual_Property > Color.
Color > Red.
sense("red", [Object]-colr->[Red]).
```

(4)
```
?- sense("red", g). ?- sense("red", g).
```

```
    {g = [Object]-colr->[Red]}
?- maximalJoin([Object]-colr->[Red], [Car], g).
    {g = [Car]-colr->[Red]}
```

Prolog+CG has "sense" as a component of the grammar. A sense graph shows what concepts are necessary to represent the meaning of a word. Although Prolog+CG introduce "sense" in order to represent the type definition of a noun, in this study, we will use "sense" in order to represent the canonical graph of an adjective as well as a noun.

As seen in (4) above, when the user requests the sense of *red*, the system searches the CG associated to "red" and then try to solve it as follows:

```
(5) ?- sense("red", g).
       {g = [Object]-colr->[Red]}
```

Then, as seen in (6) below, a maximalJoin operation between the graph [Object]-colr->[Red] and concept [Car] is requested on the interface console.

```
(6) ?- maximalJoin([Object]-colr->[Red], [Car], g).
       {g = [Car]-colr->[Red]}
```

According to the type hierarchies described in (3) above, since [Object] is a supertype of [Car], the two concepts can be joined maximally to be [Car]. Thus, the maximalJoin between [Object]-colr->[Red] and [Car] results in [Car]->colr->[Red].

Finally, this graph *g* can be added into the original program through the expert system query as in (7).

```
(7) ?- [Car]-colr->[Red].
         ==> Is it true that : [Car]-colr->[Red] ?
               type y (for yes) or n (for no) : y
```

In this way, the meaning of the combination of *red* and *car* can be obtained. Similarly, as can be seen in (8) and (9) below, the meaning of *wooden cross* can be obtained by maximalJoin between the sense graph for *wooden* and the concept of *cross*:

```
(8) ?- sense("wooden", g).
        {g = [Artifact]-matr->[Wood]}
(9) ?- maximalJoin([Artifact]-matr->[Wood], [Cross], g).
        {g = [Cross]-matr->[Wood]}
```

However, some adjective-noun combinations are ambiguous. In order to account for these ambiguities, this study suggests that an adjective can be represented through different canonical graphs, each of which includes concepts that can be linked to the given sense of the adjective.

Let us look at *criminal lawyer*, which is ambiguous in two ways: *criminal* could mean dealing with crime, or it could mean who commits a crime. These two senses of *criminal* are shown in (10) below.

```
(10) a. sense("criminal", [Conduct] -
                                    -agnt->[Person],
                                    -thme->[Lawsuit]-attr->[Crime]).
     b. sense("criminal", [Commit] -
                                    -thme->[Crime],
                                    -agnt->[Person]).
```

(10a) can be read "the agent of Conduct is Person, the theme of the Conduct is Lawsuit, and the attribute of the Lawsuit is Crime."

When the graph in (10a) and concept [Lawyer] are joined maximally, the following graph *g* will be resulted.

```
(11) ?- maximalJoin([Conduct] -
                    -agnt->[Person],
                    -thme->[Lawsuit]-attr->[Crime];, [Lawyer], g).
        {g = [Conduct] -
                    -agnt->[Lawyer],
```

-thme->[Lawsuit]-attr->[Crime]}

The graph g in (11) means *a lawyer who conducts criminal lawsuits*.

However, if the sense graph of *criminal* in (10b) and concept [Lawyer] are joined maximally, the graph *g* in (12) will be resulted. It means *a lawyer who commits a crime*.

```
(12) ?-  maximalJoin([Commit] -
                         -thme->[Crime],
                         -agnt->[Person];, [Lawyer], g).
      {g = [Commit] -
                    -thme->[Crime],
                    -agnt->[Lawyer]}
```

The adjective *musical* is also ambiguous. Let us consider *musical instrument* and *musical child*. The adjective *musical* may mean "characterized by or capable of producing music" or "talented in or devoted to music." These two senses are represented as follows:

```
(13) a. sense("musical", [Produce] -
                         -thme->[Music],
                         -inst->[Object]).
     b. sense("musical", [Person]-poss->[Talent]-attr->[Music]).
```

When the graph in (13a) and the concept for *instrument* are joined maximally, the result will look like:

```
(14)  [Produce] -
              -thme->[Music],
              -inst->[Instrument].
```

On the other hand, the maximalJoin between the graph in (13b) and the concept for *child* will produce the following graph.

```
(15)  [Child]-poss->[Talent]-attr->[Music].
```

However, what if a maximalJoin between the graph in (13a) and the concept for *child*, or a maximalJoin between the graph in (13b) and the concept for *instrument* is requested? As we can see in (16a) and (16b) below, these joins are automatically banned, since [Child] cannot be maximally joined with any concepts in (13a) (*i.e.*, [Produce], [Music], or [Object]) and [Instrument] cannot be maximally joined with any concepts in (13b) (*i.e.*, [Person], [Talent], or [Music]).

(16) a. ?- maximalJoin([Produce] -
 -thme->[Music],
 -inst->[Object];, [Child], g).
 no.
 b. ?- maximalJoin([Person]-poss->[Talent]-attr->[Music],
 [Instrument], g).
 no.

So far, we have seen cases that the meaning of an adjective-noun combination can be revealed through the join between the sense graph for the adjective and the concept for the noun. However, when it comes to such cases as *just ruler*, things are different. In order to understand its meaning, the specification of the meaning of *ruler* is required.

(17) a. sense("just", [Act]-manr->[Justice]).
 b. sense("ruler", [Person]<-agnt-[Rule]).

The maximalJoin between the sense graph for *just* and that for *ruler* yields the graph as follows.

(18) [Person]<-agnt-[Rule]-manr->[Justice]

Let us now look at *beautiful dancer*. The sentence *Mary is a beautiful dancer* has two different meanings: *Mary is beautiful as a dancer* or *Mary is beautiful and a dancer*. The two graphs in (19) show the two senses of *beautiful*.

(19) a. sense("beautiful",
[Beauty]<-char-[Appearance]<-poss-[Person]).
b. sense("beautiful", [Act]-manr->[Beauty]).

The maximalJoin between the graph in (19a) and [Dancer] yields a graph in (20) below, where [Dancer] is the result of the maximalJoin between [Person] and [Dancer] since Person can be merged into its subtype Dancer.

(20) [Beauty]<-char-[Appearance]<-poss-[Dancer]

On the other hand, the graph in (19b) cannot be combined with [Dancer], since Dancer cannot be maximally joined with [Act] or [Beauty]. Thus, for the graph in (19b) to be joined with [Dancer], the sense graph for *dancer* should be introduced into the program as shown in (21) below.

(21) sense("dancer", [Dance]-agnt->[Person]).

After the graph for *dancer* is retrieved, the graph in (19b) and the graph in (21) can be maximally joined, resulting in the graph (22) as follows:

(22) [Dance] -
-manr->[Beauty],
-agnt->[Person].

If the graph for *beautiful* in (19a) and the graph for *dancer* in (21) are joined, then the result will be (23), which has the same meaning as (20) above.

(23) [Beauty]<-char-[Appearance]<-poss-[Person]-agnt->[Dance].

Now, let us consider the adjective *good*. *Good mechanic* has two interpretations in accordance with the two senses of *good.* (24a) says that the quality of the skill is good, while (24b) means that the attitude is good.

(24) a. sense("good", [Act]-qual->[Superiority]).
     b. sense("good", [Animate]-poss->[Attitude]-attr->[Goodness]).
(25) sense("mechanic", [Person]<-agnt-[Repair]-obj->[Machine]).
(26) a. [Repair] -
          -qual->[Superiority],
          -agnt->[Person],
          -obj->[Machine].
     b. [Repair] -
          -agnt->[Person]-poss->[Attitude]-attr->[Goodness],
          -obj->[Machine].

The similar procedure can be found in such cases as *poor teacher/liar/linguist*. The program has two different graphs for *poor* (*i.e.,* the economic status vs. the quality of a skill) and the graphs for the nouns such as *teacher, linguist,* and *liar*.

(27) a. sense("poor", [Person]-poss->[Possession]-amount->[Few]).
     b. sense("poor", [Act]-qual->[Inferiority]).
(28) a. sense("teacher",
          [Class_subject]<-thme-[Teach]-agnt->[Person]).
     b. sense("linguist", [Study] -
          -agnt->[Person],
          -thme->[Linguistics]).
     c. sense("liar", [Lie]-agnt->[Person]).

Each of the two graphs for *poor* in (27) can be combined with each of the graphs for *teacher, linguist,* and *liar* in (28). For instance, for *poor teacher,* two different maximalJoin operations can be requested, resulting in two different graphs as in (29) and (30).

(29) ?-maximalJoin([Person]-poss->[Possession]-amount->[Few],

[Class_subject]<-thme-[Teach]-agnt->[Person], g).
{g = [Teach] -
-thme->[Class_subject],
-agnt->[Person]-poss->[Possession]-amount->[Few]}
(30) ?- maximalJoin([Act] -
-agnt->[Person],
-qual->[Inferiority];, [Teach] -
-agnt->[Person],
-thme->[Class_subject];, g).
{g = [Teach] -
-agnt->[Person],
-qual->[Inferiority],
-thme->[Class_subject]}

In a similar manner, the meaning for *poor linguist* and *poor liar* will be revealed as follows: (31a) and (31b) are for *poor linguist,* and (32a) and (31b) for *poor liar*.

(31) a. [Study] -
-agnt->[Person]-poss->[Possession]-amount->[Few],
-thme->[Linguistics].
b. [Study] -
-agnt->[Person],
-qual->[Inferiority],
-thme->[Linguistics].
(32) a. [Lie]-agnt->[Person]-poss->[Possession]-amount->[Few].
b. [Lie] -
-agnt->[Person],
-qual->[Inferiority].

Noting that these adjectives like *beautiful, good,* and *poor* can modify an implicit event of a kind appropriate for the object denoted by the nominal, Larson (2000) states that without invocation of an independent, implicit event there would be a very serious compositionality problem. However, he does not specify what 'an implicit event of a kind appropriate' for the object is.

The virtue of the approach presented in this study is that it can explicitly reveal the appropriate events that adjectives modify since the senses of nouns as well as those of adjectives can be declared in the program.

Now, let us consider *English teacher*, which has two readings (*i.e.*, 'person who teaches English' and 'person who teaches and who is English'). We have two sense graphs for *English* in (33) and a sense graph for *teacher* in (34) as follows:

(33) a. sense("English", [Person]-stat->[Born]-place->[England]).
b. sense("English", [Language]-attr->[English]).
(34) sense("teacher", [Person]<-agnt-[Teach]-thme->[Class_subject]).

The graph in (34) can be combined with (33a) or (33b), yielding two maximally-joined graphs in (35a) and (35b) respectively.

a. [Teach] -
-agnt->[Person],
-thme->[Language]-attr->[English].
b. [Teach] -
-agnt->[Person]-stat->[Born]-place->[England],
-thme->[Class_subject].

(35)

Now, let us take a look at *old rival*. Larson (2000) presents that *old* displays an apparent three-way ambiguity: "aged", "longstanding", and "former."

(36) Max is an old rival.
a. "an aged rival" Max is a rival and he is old.
b. "a longstanding rival" The rivalry with Max is old.
c. "a former rival" Max was a rival, but he is no longer one.

For the meaning of *old rival*, three graphs for *old* can be given as seen in (37) below. Each of the graphs in (37a), (37b), and (37c) corresponds to (36a), (36b), and (36c) respectively. The sense graph for *rival* can be structured as in (38).

(37) a. sense("old", [Live] -
-dur->[Long],
-agnt->[Person]).
b. sense("old", [Act]-dur->[Long]).
c. sense("old", [Act]-ptim->[Time]-succ->[Present]).
(38) sense("rival", [Defeat] -
-ptnt->[Contestant],
-agnt->[Person]).

Then, the maximalJoins are requested between each graph for *old* and the graph for *rival,* yielding three different graphs as in (39).

(39) a. ?-maximalJoin([Person]<-agnt-[Live]-dur->[Long], [Rival], g).
{g = [Live] -
-agnt->[Rival],
-dur->[Long]}
b. ?- maximalJoin([Act]-dur->[Long],
[Person]<-agnt-[Defeat]-ptnt->[Contestant], g).
{g = [Defeat] -
-dur->[Long],
-agnt->[Person],
-ptnt->[Contestant]}
c. ?- maximalJoin([Act]-ptim->[Time]-succ->[Present],
[Person]<-agnt-[Defeat]-ptnt->[Contestant], g).
{g = [Defeat] -
-ptim->[Time]-succ->[Present],
-agnt->[Person],
-ptnt->[Contestant]}

(39c) conveys the 'former' reading. Now, let us consider the temporal adjective *former* as in *former president*. *Former* must be represented exactly the same as the third sense of *old* mentioned in (37c) above. The graph for *former* and the graph for *president* will be represented as follows.

(40) sense("former", [Act]-ptim->[Time]-succ->[Present]).
(41) sense("president",

[Person]<-agnt-[Administer]-obj->[Government]).

The maximalJoin between the two graphs in (40) and (41) will yield the graph in (42).

(42) [Administer] -
-ptim->[Time]-succ->[Present],
-agnt->[Person],
-obj->[Government].

In this way, the meaning of adjective-noun combinations such as *beautiful dancer, poor linguist, etc.* can be accounted for through sense graphs of the nouns. However, adjectives and nouns can be combined in a more indirect way through collections of schemata (*i.e.*, world knowledge) for nouns. The possible examples of this type include *recent letter* and *quick cup of coffee*. Even though neither *letter* nor *cup* appears to be an eventive noun, letters readily invoke "surrounding events" of writing, sending, receiving and reading.

Violi (2001) mentions that knowledge is structured on the basis of organized sets of concepts that are systematically linked to each other in schemata, and, thus, these notations can be extremely helpful for semantic representation and in general for a theory of lexical meaning.

In CGs, the schema is the basic structure for representing background knowledge for human-like inference. The schema describes the conventional, normally occurring, or default roles that a concept plays with respect to other concepts. By joining schemata, the inference engine expands a query graph to a working graph that incorporates additional background information.

For *recent letter*, we may have two different schema graphs as given in (43): one related to the event of writing and the other related to the event of receiving.

(43) a. Letter(schema)::[Write] -
-agnt->[Sender],
-rslt->[Letter]<-obj-[Read]-agnt->[Receiver].

b. Letter(schema)::[Receive] -
-agnt->[Receiver],
-obj->[Letter]<-rslt-[Write]-agnt->[Sender].

If *a recent letter* means a letter recently written, then the schema graph in (43a) will be joined with the sense graph for *recent* (*i.e.*, sense("recent", [Act]-ptim->[Time]-char->[Close]<-char-[Present]), yielding the following graph:

(44) [Write] -
-ptim->[Time]-char->[Close]<-char-[Present],
-agnt->[Sender],
-rslt->[Letter]<-obj-[Read]-agnt->[Receiver].

If *a recent letter* means a letter recently received, then the schema graph in (43b) and the sense graph for *recent* will be joined maximally, resulting in the following graph:

(45) [Receive] -
-ptim->[Time]-char->[Close]<-char-[Present],
-agnt->[Receiver],
-obj->[Letter]<-rslt-[Write]-agnt->[Sender].

Similarly, in order to interpret *quick cup of coffee*, a schema for *coffee* should be retrieved from the program as well as the sense graph for *quick*. If we consider a term like *coffee*, we find that its semantic representation includes a wide range of encyclopedic properties. We know that coffee is an object of drinking, it is dark, it has a particular flavor, it has stimulating effects because it contains caffeine, and so on.

(46) ?- sense("quick", g).
{g = [Act]-manr->[Quick]}
(47) ?- Coffee(schema):: G.
{G = [Coffee] -

<-obj-[Drink]-inst->[Cup],
-cont->[Caffeine],
-poss->[Flavor]}

The schema in (47) shows that, among other things, [Coffee] is related to Act [Drink] and, in turn, [Drink] has an instrument relation with [Cup]. Thus, these links allow *quick* and *cup of coffee* to be joined maximally. The resulted graph shown in (48) below says that the object of [Drink] is [Coffee] and the manner of [Drink] is [Quick].

(48) [Coffee] -
<-obj-[Drink] -
-manr->[Quick],
-inst->[Cup];,
-cont->[Caffeine],
-poss->[Flavor].

Now, let us consider the "celebrated" example *fake gun*. There have been different approaches to a class of concept combination known as "privatives". Kamp (1975) defines privative adjectives as adjectives such that, given adjective A and noun N, the claim 'No AN is a N' is necessarily true.

However, Partee (2001) proposes that adjectives *fake* and *imaginary* are not actually privative, but subsective, and that no adjectives are actually privative. She hypothesizes that in interpreting a sentence like *I don't care whether that fur is fake fur or real fur*, we actually expand the denotation of 'fur' to include both fake and real fur. Frank (1995) also says that if *x is* a *fake gun*, then *x is* a *gun* in some sense and he suggests the sense generation model by showing how to handle *fake gun*. He mentions Fires(Bullets), Made-of(Metal), and Function(Kill) as central features of the concept gun, and Trigger(+), Barrel(+), and Handle(+) as diagnostic features.

On the other hand, Coulson and Fauconnier (1999) and Coulson (2001) suggest that the character of a *fake gun* will depend on the faker's motivation, the scenario in which the gun is to function, and the knowledge of the prospective victim. They also say that the characteristics of fake objects arise because of the way in which the intent to deceive is central to the concept of *fake*, and that the blending process enables speakers to make connections between elements whose objective properties may be materially different.

Within the CG framework, this study integrates the sense generation model and the conceptual blending model, suggesting that in the combination of *fake gun*, *gun* has a schema which includes its material, parts, and functions. The sense graph for *fake* should include both the intent to deceive and the fake object which is believed to be a real one.

The schema for *gun* is shown in (49), and the sense graph for *fake* is given in (50), which has the multi-referent notation to specify several occurrences of the same concept: the multi-referent *1 is used to specify that the two concept of [Object: *1] are in fact two occurrence of the same concept.

```
(49)  Gun(schema)::[Gun] -
                          -matr->[Metal],
                          -part->[Trigger],
                          -part->[Barrel],
                          -part->[Handle],
                          -poss->[Bullet],
                          -purp->[Kill].
(50)   sense("fake", [Person]<-agnt-[Deceive]-
              -ptnt->[Person]-stat->[Believe]-thme->
                    [Object]-stat->[Identity]<-stat-[Object :*1],
              -inst->[Object]).
```

When these two graphs are joined maximally, the graph in (51) is resulted.

```
(51) [Deceive] -
              -agnt->[Person],
              -ptnt->[Person]-stat->[Believe]-thme->[Gun] -
                       -stat->[Identity]<-stat-[Object : *1],
                       -matr->[Metal],
```

```
-part->[Trigger],
-part->[Barrel],
-part->[Handle],
-poss->[Bullet],
-purp->[Kill];,
-inst->[Object : *1].
```

The graph in (51) says that an object is the instrument of deception and someone is deceived into believing that the object is a gun which has those properties mentioned in (49). But since the instrument has the conceptual type of Object, it can refer to anything whose conceptual type is a subtype of Object. That is, it might be a pencil, a finger, or even a zucchini. Since, in the schema graph, the gun has the information about the material, the function, and parts of a gun, the graph *g* in (51) shows the general knowledge about a gun. Thus, the graph shows not merely Coulson and Fauconnier's viewpoint (*i.e.*, the actor's intention to deceive and the victim's would-be-belief), but Frank's (*i.e.*, the fact that there seem to be basic gun features (e.g., being a weapon (Gun is a subtype of Weapon), having a barrel, theoretical ability to shoot bullets, and so forth) that cannot be altered in a context such as a robbery or even in a game context such as a cops and robbers game).

Now, let us take a look at the case of *tall person*. The semantic contribution of adjectives such as *tall* and *heavy* is dependent on the head nouns that they modify. *Tall* denotes one range of heights for a person, another for a tree, still another for a building. It is likely that part of the meaning of each of the nouns is a range of expected values for the attribute HEIGHT. *Tall* is interpreted relative to the expected height of objects of the kind denoted by the head noun. The graph in (52) represents that the height is superior to the average.

```
(52) sense("tall",
        [Average]-sup->[Measure]<-mesr-[Height]<-attr-[Thing]).
```

The default average will be retrieved later when the sense graph of *tall* is combined with the schema of a noun which follows the adjective. For instance, as seen in (53) below, the schema of *person* may include Height whose default height might be 170cm.

(53) Person(schema)::[Person] -
-attr->[Height] -
-mesr->[Measure],
-char->[Average = 170];,
-attr->[Weight].

In this case, the maximalJoin between the graphs in (52) and (53) will produce the graph as follows:

(54) [Person] -
-attr->[Height] -
-mesr->[Measure]<-sup-[Average],
-char->[Average = 170];,
-attr->[Weight].

The graph in (54) says that a person has attributes such as height and weight, and the measure (*i.e.*, value) of his or her height is superior to the average of 170cm.

4. Conclusion

This study has dealt with the semantic representations of English adjective-noun combinations, which are implemented in Prolog+CG. This study suggests that in order to account for the meaning of adjective-noun combinations, the type definitions of the concepts for the adjectives should be declared. An adjective should not be mapped into a single concept, as in the case of a noun, but rather should be mapped into a canonical graph containing other concepts that consist of the meaning of the adjective. This study also shows that there are cases where in addition to the type definitions of the adjectives, the type definitions and the schemata of the concepts for the nouns should be specified in order to understand the meaning of adjective-noun combinations

This paper has shown that the meaning of adjective-noun combinations can be revealed in three ways. The first type is demonstrated in the cases where the meaning of adjective-noun combinations can be specified through the maximalJoin between the sense graph for the adjective and the concept for the noun.

The second type is examplified in the cases where the meaning can be obtained by means of the maximalJoin between the sense of graph for the adjective and the sense graph for the noun.

The third type is revealed in the cases where schematic knowledge about the noun is related: since a sense graph for an adjective has a concept which is included in the schema graph for a noun, the meaning of the combination can be obtained through the maximalJoin between the two graphs.

References

Bouillon, P & Viegas, E. (1999). The description of adjectives for natural language processing: Theoretical and applied perspectives. *TALN 1999*.

Coulson, S. (2001). *Semantic leaps: Frame-shifting and conceptual blending in meaning construction*. Cambridge: Cambridge University Press.

Coulson, S. & Fauconnier, G. (1999). Fake gun and stone lion: Conceptual blending and privative adjectives. In B. Fox, D. Jurafsky, & L. Michaelis (Eds.), *Cognition and Function in Language* (pp. 143-158). Stanford, CA: CSLI Publications.

Fellbaum, C., Gross, D., & Miller, K. (1993). Adjectives in WordNet. In G. A. Miller, R. Beckwith, & C. Fellbaum (Eds.), Five papers on WordNet. *Cognitive science laboratory Report, 43*.

Frank, B. (1995). Sense generation: A "quasi-classical" approach to concepts and concept combination. *Cognitive science, 19,* 441-505.

Jung, M. A. (2002). A Semantic-network approach to indirect anaphora resolution in English. *The Journal of Studies in Language, 18*(1), 139-157.

Kabbaj, A., Moulin, B., Gancer, J., Nadeau, D., & Rouleau, O. (2001). Uses, improvements, and extensions of Prolog+CG: Case studies. In H. S. Delugach & G. Stumme (Eds.), *Conceptual structures: broadening the base. 9th International Conference on Conceptual Structures, ICCS 2001. Stanford, CA, USA, July 30-August 3, 2001 Proceedings*. (pp. 346-359). Berlin, Germany: Springer.

Kamp, H. (1975). Two theories about adjectives. In E. L. Keenan (Ed.), *Formal Semantics of Natural Language*. Cambridge: Cambridge University Press.

Miller, K. J. (1998). Modifiers in WordNet. In C. Fellbaum (Ed.), *WordNet: An electronic lexical database*. Cambridge, MA: The MIT Press.

Larson, R. K. (2000). Semantics of adjectival modification. Unpublished manuscript. *Lectures at the LOT Winter School*.

Partee, B. H. (2001). Privative adjectives: Subsective plus coercion. To appear in T. E. Zimmerman, (Ed.), *Studies in presupposition*.

Raskin, V. & Nirenburg, S. (1995). Lexical semantics of adjectives: A microtheory of adjective meaning. *MCCS-95-288*.

Schärfe, H. (2002). Searching for Narrative Structures. In Harabagiu, S. & Chaudhri, V. (Eds.) *Mining Answers from Texts and Knowledge Bases*. Papers from the 2002 AAAI Symposium. March 25-27, Stanford California. Technical Report SS-02-06, AAAI Press, Menlo Park.

Sowa, J. F. (1984). *Conceptual structures: Information processing in mind and machine*. California: Addison-Wesley Publishing Co., Inc.

Sowa, J. F. (1991). Toward the expressive power of natural language. In J. F. Sowa (Ed.), *Principles of semantic networks: Explorations in the representation of knowledge* (pp. 157-189). San Mateo, CA: Morgan Kaufmann Publishers.

Sowa, J. F. (1992). Conceptual graphs summary. In Nagle, Nagle, Gerholz, & Eklund, (Eds.) *Conceptual Structures: Current research and practice*. Ellis Horwood Workshops, 1-51.

Sowa, J. F. (1999). Controlled English. Retrieved February 24, 2002, from http://users.bestweb.net/~sowa/misc/ace.htm

Sowa, J. F. (2000). *Knowledge representation: Logic, philosophical, and computational foundations*. Boston: PWS Publishing Co.

Viegas, E. (1998). Multilinguality and genericity in syntagmatic relations. In *Proceedings of the Second International Workshop on Multilinguality in the Lexicon,* as part of *the 13th European Conference in Artificial Intelligence*.

Viegas, E., Beale, S., & Nirenburg, S. (1998). The computational lexical semantics of syntagmatic relations. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*.

Violi, P. (2001). *Meaning and experience*. Bloomington, IN: Indiana University Press.

MieAe Jung
Computer Science Department
Indiana University
Bloomington, IN 47405, USA
Phone: 1-812-857-3238
Email: majung@indiana.edu

Received in July, 2002
Accepted in November, 2002