Language-Neutral Representation of Syntactic Structure

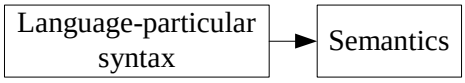
Richard Campbell and Hisami Suzuki
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
{richcamp, hisamis}@microsoft.com

Abstract

We propose a semantically motivated linguistic representation called Language-Neutral Syntax (LNS), which is scalable in three important respects. First, though information stored in LNS is directly related to the surface tree, it is abstract enough to normalize many surface variations both within and across languages. Second, LNS is adaptable to new applications, in that any application that requires a particular kind of semantic information can extract that information from LNS. Finally, since LNS is developed as part of a broad-coverage parser, it is designed to handle a wide range of constructions. LNS is currently implemented in a large-scale, multi-lingual natural language understanding system, and is used in applications requiring various kinds of semantic information, including question answering and machine translation.

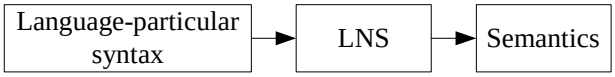
1 Introduction

Systems for natural language understanding typically make use of some kind of semantic or quasi-semantic representation which is derived from a surface-based syntactic analysis; this is diagrammed below:



The surface syntax analysis is perforce particular to individual languages, since languages vary widely in constituent order, morphosyntax, etc.

In this paper we propose a semantically motivated syntactic representation we refer to as Language-Neutral Syntax (LNS); LNS differs from surface-based syntactic representations in that it abstracts away from language-particular properties of the structure of sentences such as constituent order and morphosyntax, and represents what might be thought of as a sentence's underlying, or logical, syntax. LNS thus fits in between language-particular surface syntax and semantics:



LNS is neither a comprehensive semantic representation, nor a syntactic analysis of a particular language, but a semantically motivated, language-neutral syntactic representation. This conception of LNS provides three main advantages:

- *Scalability across languages*: LNS is not specific to any language, but is used as a shared representation by typologically diverse languages; e.g. sentential negation in English and Japanese have the same representation (Section 3).
- *Scalability across applications*: LNS nevertheless remains fairly close to surface syntax, insofar as information that is present in the surface structure is retained; thus it is not particular to any application, but can be adapted to new applications as needed (Section 4).
- *Allows broad coverage*: By avoiding full-fledged semantic analysis, LNS can accommodate a broad range of actually-occurring constructions (Section 5).

The combination of these three points, we believe, is what primarily distinguishes LNS from other frameworks; this will be elaborated in Section 5.

2 Overview of LNS

2.1 Input to LNS

The input to LNS can be any surface-syntactic annotation produced by a parser or manually; currently, LNS is implemented in the NLPWin[1] system being developed at Microsoft Research.

[1] As described in Heidorn (2000); NLPWin currently

The examples in (1) – (3) below are sentences in English (1, 2) and Japanese (3), all of which mean roughly the same thing:

(1) The cat chased the mouse.
(2) The mouse was chased by the cat.
(3) ねずみがねこに追いかけられた。
nezumi-ga neko-ni oikake-rare-ta
mouse-NOM cat-DAT chase-PASS-PAST

There is a good deal of surface variability among these sentences: the active (1) and passive (2) constructions are quite different in its constituent structure, let alone the equivalent sentence of (2) in Japanese (3).

The goal of LNS is to provide a representational framework which is sufficiently abstract to normalize surface variability both within and across languages, while at the same time retaining all significant information in the surface structure.

2.2 Description of LNS

An LNS representation is like a standard analysis tree in two important respects: (i) it is an annotated tree, in which each non-root node has exactly one parent; and (ii) it supports a notion of constituency, insofar as the tree contains non-terminal nodes (labeled either FORMULA or NOMINAL). LNS differs from a standard analysis tree, however, in two ways: (i) the immediate constituents of an LNS node are not ordered; (ii) in place of surface relations, LNS expresses semantically motivated relations more transparently, such as deep grammatical functions and scope of modifiers and operators.

This is best illustrated by example. Consider (4) – (6), the LNSs for sentences (1) – (3) respectively:[2]

(4) The cat chased the mouse.

```
FORMULA1 (+Past +Proposition)
|_L_Sub---NOMINAL1 (+Def +Sing)
          |_SemHeads--cat1
|_L_Obj---NOMINAL2 (+Def +Sing)
          |_SemHeads--mouse1
|_SemHeads--chase1
```

(5) The mouse was chased by the cat.

```
FORMULA1 (+Past +Pass +Proposition)
|_L_Sub---NOMINAL1 (+Def +Sing)
          |_SemHeads--cat1
|_L_Obj---NOMINAL2 (+Def +Sing)
          |_SemHeads--mouse1
|_SemHeads--chase1
```

(6) ねずみがねこに追いかけられた。

```
FORMULA1 (+Past +Pass +Proposition)
|_L_Sub---NOMINAL1
          |_SemHeads--ねこ1 (cat)
|_L_Obj---NOMINAL2
          |_SemHeads--ねずみ1 (mouse)
|_SemHeads--追い掛ける1 (chase)
```

We can see that in LNS, the information that is in the surface structure is retained, albeit in a different form. For example, each LNS records the basic arguments of the clause explicitly (such as logical subject (*L_Sub*), logical object (*L_Obj*) and semantic head (*SemHeads*)), while such information is only implicit in the surface structure, and is encoded by language-particular devices (such as by word order in English and by agglutinative morphology in Japanese). Also, LNS normalizes syntactic and morphological variation: passive and active pairs such as (1) and (2) have a neutralized argument structure (though the information that (2) is passive is retained in (5)); likewise, morphological passive is normalized in (6). Individual words are also lemmatized.

LNS is thus structurally language-neutral: (5) and (6) are the same, aside from lexical items (which are perforce language-specific; LNS is not an interlingua), despite the differences in their surface realizations in (2) and (3). Put another way, the grammatical differences between (5) and (6) are due to arbitrary, language-specific, *surface* syntax conventions; LNS represents their identical *logical* syntax, as it were, and is thus the same for both.

Another example that illustrates the language-neutrality of LNS well is the causative construction:

(7) He had the cat eat.

```
FORMULA1 (+Past +Proposition)
|_L_Sub---NOMINAL1
          |_SemHeads--he1
|_L_Obj---FORMULA2
          |_L_Sub---NOMINAL2 (+Def)
                    |_SemHeads--cat1
          |_SemHeads--eat1
|_SemHeads--have1
```

(8) 彼はねこに食べさせた。
kare-wa neko-ni tabe-sase-ta
he-TOP cat-DAT eat-CAUS-PAST

```
FORMULA1 (+Past +Proposition)
|_L_Sub---NOMINAL1
          |_SemHeads--彼1 (he)
|_L_Obj---FORMULA2
          |_L_Sub---NOMINAL2
                    |_SemHeads--ねこ1 (cat)
          |_SemHeads--食べる1 (eat)
|_SemHeads--させる1 (cause)
```

---

includes full-fledged parsers in seven languages (Chinese, English, French, German, Japanese, Korean, Spanish), each with a lexicon of 70,000 to 120,000 words.

[2] To save space we display only those features relevant to the present discussion.

The structure of the English causative in (7) is biclausal on the surface, as it is in LNS, where FORMULA2 is the logical object (i.e., complement) of FORMULA1. Japanese (8), on the other hand, is monoclausal on the surface: the causative is represented morphologically in the single verb *tabe-sase-ta* 'eat-CAUS-PAST'. However, it has a biclausal structure in LNS, which is not only motivated by Japanese-internal linguistic evidence (see e.g., Shibatani, 1990), but also contributes to normalizing grammatical constructions cross-linguistically in LNS.

Non-local dependencies caused by long-distance displacement of constituents are also normalized in LNS. For example, the fronted wh-phrase in (9) is in its underlying argument position, as L_Obj of FORMULA2:

(9) I know who you saw.

```
FORMULA1 (+Pres +Proposition)
|_L_Sub---NOMINAL1
          |_SemHeads--I1
|_L_Obj---FORMULA2 (+Past +WhQ)
          |_L_Sub---NOMINAL2
                    |_SemHeads--you1
          |_L_Obj---NOMINAL3 (+Rel )
                    |_SemHeads--who1
          |_SemHeads--see1
|_SemHeads--know1
```

The surface position of the wh-phrase is indirectly captured in this case by the +WhQ feature on FORMULA2, which indicates that FORMULA2 is interpreted as a wh-question, and thus marks the scope of the wh-operator.

Obligatory coreference is represented by a control relation, mediated by the *Cntrlr* attribute. This occurs in Equi constructions, relative clauses, and others; (10) shows an NP with a relative clause, with the Cntrlr attribute pointing to NOMINAL4, where *L_Attrib* (logical attribute) indicates the general relation for attributive modifiers.

(10) the small boy that I met

```
NOMINAL1 (+Def)
|_L_Attrib--FORMULA1 (+Past +Proposition)
          |_L_Sub---NOMINAL2
                    |_SemHeads--I1
          |_L_Obj---NOMINAL3 (Cntrlr: NOMINAL4)
                    |_SemHeads--that1
          |_SemHeads--meet1
|_SemHeads--NOMINAL4
          |_L_Attrib--FORMULA2
                    |_SemHeads--small1
          |_SemHeads--boy1
```

3 Scalability across languages

Our claim is that LNS is scalable to new languages; i.e., it is a language-neutral representation, in the sense that arbitrary, language-specific conventions for the realization of various semantic notions are normalized into a single, coherent, and semantically motivated structure. In this section, we illustrate this concept by looking at modifier scope within NP and sentential negation in some detail, two domains that show substantial cross-linguistic variation in their surface realization, which LNS normalizes to show a common logical syntax.

3.1 Cross-linguistic variation in word order

Word order carries various kinds of grammatical information that may be preserved in a more useful manner in LNS. For example, it contributes to the identification of grammatical relations (e.g., Subject, Object), topic/comment structure, and scope relations among modifiers and logical operators. We have seen that grammatical relations are expressed via attributes such as L_Sub in LNS. This section is concerned with the language-neutral representation of scope relations among modifiers in LNS.[3]

Consider first the English NP *the heaviest natural isotope*, where the order of the adjectives reflects their logical scope: *heaviest* modifies *natural isotope*, so the NP refers to the heaviest member of the set of natural isotopes, and not necessarily to the heaviest isotope overall (there may be a heavier one that is synthetic). The scope of these modifiers surfaces in English as left-to-right order; but it may surface differently in another language. For example, one translation of this NP into French is *l'isotope naturel le plus lourd*, lit. 'the isotope natural the most heavy'; though the linear order of modifiers is the opposite of English, their logical scope, and hence their LNS, is the same.

(11) the heaviest natural isotope

```
NOMINAL1 (+Def)
|_L_Attrib--FORMULA1 (+Supr)
          |_SemHeads--heavy1
|_SemHeads--NOMINAL2
          |_L_Attrib--FORMULA2
                    |_SemHeads--natural1
          |_SemHeads--isotope1
```

(12) l'isotope naturel le plus lourd
the isotope natural the most heavy

```
NOMINAL1 (+Def)
|_L_Attrib--FORMULA1 (+Supr)
          |_SemHeads--lourd1
|_SemHeads--NOMINAL2
          |_L_Attrib--FORMULA2
                    |_SemHeads--naturel1
          |_SemHeads--isotope1
```

[3] The computation of modifier scope is not discussed here; see Campbell (2002) for a discussion of scope computation.

In both cases, the superlative modifier modifies the constituent containing the simple attributive adjective (and thus has wider scope). The two languages realize the relative scope of the adjectives differently on the surface, but LNS normalizes this difference into a uniform representation.

As another example, consider the NP *the first perfume that used alcohol* (13) and its Japanese translation (14) *arukooru-o mochii-ta saisho-no kousui*, lit. 'alcohol-ACC use-PAST first-ADN perfume'; in this case, word order is merely conventional, and has no semantic significance: In English, relative clauses must be postnominal and unmodified adjectives prenominal, regardless of their relative logical scope; in Japanese, relative clauses typically precede other prenominal modifiers, again regardless of scope. In LNS, these are normalized to a representation that shows the relative logical scope of the modifiers, ignoring arbitrary ordering conventions:

(13) the first perfume that used alcohol

```
NOMINAL1 (+Def)
|_L_Attrib--FORMULA1 (+Supr)
              |_SemHeads--first1
|_SemHeads--NOMINAL2
              |_L_Attrib--FORMULA2 (+Past)
                            |_L_Sub---NOMINAL3 (+Rel)
                                          |_SemHeads--that1
                            |_L_Obj---NOMINAL4
                                          |_SemHeads--alcohol1
                            |_SemHeads--use1
              |_SemHeads--perfume1
```

(14) アルコールをもちいた最初の香水
arukooru-o mochii-ta saisho-no kousui
alcohol-ACC use-PAST first-ADN perfume

```
NOMINAL1
|_L_Attrib--FORMULA1 (+Supr)
            |_SemHeads--最初1 (first)
|_SemHeads--NOMINAL2
              |_L_Attrib--FORMULA2 (+Past)
                            |_L_Sub---NOMINAL3 (+Rel)
                                          |_SemHeads--の1 (that)
                            |_L_Obj---NOMINAL4
                                          |_SemHeads--アルコール1
                                                        (alcohol)
                            |_SemHeads--もちいる1 (use)
                |_SemHeads--香水1 (perfume)
```

Again, although the two languages realize the scope of these modifiers differently on the surface, they have the same LNS.

3.2 Negation

3.2.1 Representing negation in LNS

Sentential negation is indicated by a negative operator, which is the semantic head in LNS. The scope of the operator (OpDomain) is the LNS constituent corresponding to the sentence without negation:

(15) He didn't die.

```
FORMULA1 (+Past +Proposition)
|_OpDomain--FORMULA2
                |_L_Sub---NOMINAL1
                              |_SemHeads--he1
                |_SemHeads--die1
|_SemHeads--not1
```

A negative operator need not be a lexical item; in Japanese, for instance, negation is always expressed via verbal or adjectival inflection. In these cases, LNS has an abstract *_NEG* operator:

(16) 彼は死ななかった。
kare-wa sin-ana-katta
he-TOP die-NEG-PAST

```
FORMULA1
|_OpDomain--FORMULA2
                    |_L_Sub---NOMINAL1
                                  |_SemHeads--彼1 (he)
            |_SemHeads--死ぬ1 (die)
|_SemHeads--_NEG1
```

Another case where *_NEG* is required is in the case of English negative quantifiers such as *nothing*, which incorporate a negative element:

(17) I have nothing.

```
FORMULA1 (+Pres +Proposition)
|_OpDomain--FORMULA2
              |_L_Sub---NOMINAL1
                            |_SemHeads--I1
              |_L_Obj---NOMINAL2 (+ExstQuant)
                            |_SemHeads--nothing1
              |_SemHeads--have1
|_SemHeads--_NEG1
```

Besides the negative operator, (17) has a negative quantifier with the feature *+ExstQuant*, indicating that despite having the lemma of a negative quantifier, it is semantically existential. Thus (17) is interpreted as ¬[∃x[have(I,x)]]; i.e., 'it is not the case that I have something'. The +ExstQuant feature is also used for negative polarity quantifiers, such as *anybody*, for the same purpose. For illustration, compare (18) and (non-standard) (19) to (17):

(18) I don't have anything.

```
FORMULA1 (+Pres +Proposition)
|_OpDomain--FORMULA2
              |_L_Sub---NOMINAL1
                            |_SemHeads--I1
              |_L_Obj---NOMINAL2 (+ExstQuant)
                            |_SemHeads--anything1
              |_SemHeads--have1
|_SemHeads--not1
```

(19) I don't have nothing.[4]

```
FORMULA1 (+Pres +Proposition)
|_OpDomain--FORMULA2
                |_L_Sub---NOMINAL1
                                |_SemHeads--I1
                |_L_Obj---NOMINAL2 (+ExstQuant)
                                |_SemHeads--nothing1
                |_SemHeads--have1
|_SemHeads--not1
```

The only difference between (17), (18) and (19) is in the lemmas of the negative operator and the +ExstQuant quantifier. With this representation, we can express the fact that they all have the same semantic interpretation while still recording their surface differences

3.2.2 Normalizing cross-linguistic variation

Sentential negation is a good example of a domain in which languages show arbitrary, language-specific surface differences, especially as it involves negative or negative polarity quantifiers and adverbs. It is instructive to compare the translations of (17) – (19) in German, Japanese and French:

(20) Ich habe nichts.
I have nothing

```
FORMULA1 (+Pres +Proposition)
|_OpDomain--FORMULA2
                |_L_Sub---NOMINAL1
                                |_SemHeads--ich1
                |_L_Obj---NOMINAL2 (+ExstQuant)
                                |_SemHeads--nichts1
                |_SemHeads--haben1
|_SemHeads--_NEG1
```

(21) 何も持っていない。
nani-mo motte-i-nai
what-NEGPOL have-STATE-NEG
'(I) don't have anything'

```
FORMULA1 (+Pres +Proposition)
|_OpDomain--FORMULA2
        |_L_Sub---_X1
        |_L_Obj---NOMINAL1
                |_L_Quant--NOMINAL2 (+ExstQuant)
                                |_SemHeads--何も1 (any)
                |_SemHeads--_DUMMY1
        |_SemHeads--持つ1 (have)
|_SemHeads--_NEG1
```

(22) Je n' ai rien.
I not have nothing

```
FORMULA1 (+Pres +Proposition)
|_OpDomain--FORMULA2
                |_L_Sub---NOMINAL1
                                |_SemHeads--je1
                |_L_Obj---NOMINAL2 (+ExstQuant)
                                |_SemHeads--rien1
                |_SemHeads--avoir1
|_SemHeads--ne1
```

German has negative quantifiers such as *nichts* 'nothing', but there is no negative polarity counterpart to (20). Conversely, Japanese has no negative quantifiers, but allows only negative polarity quantifiers, which are always accompanied by negative inflection on the predicate, as in (21).[5] In French (22), *rien* is a negative quantifier; in standard French, a sentence containing a negative word must also have the preverbal negative particle *ne*. These variations are arbitrary, language-particular ways of realizing sentential negation, and are normalized in LNS: (17) – (22) are identical in relevant respects, reflecting their identical logical syntax.

4 Scalability across applications

Different applications require different kinds of information; rather than designing a semantic representation that is tailored to specific application needs, we have designed LNS to be flexible enough to store different kinds of information, which can be easily extracted from LNS as needed by different applications. What ensures scalability to new applications is the fact that LNS, being a syntactic representation itself, remains relatively true to surface syntax; all meaningful information that is in the surface structure of a sentence is incorporated into LNS, allowing more specific representations to be derived from LNS. In this section, we illustrate this concept with examples from question answering (QA) and machine translation (MT).

4.1 Question answering using predicate-argument structure

Lexical dependencies are not directly represented in LNS, but are implicit; the reason for this is that we view lexical dependencies as a relatively deep semantic notion, a step removed in abstraction from the logical syntax of a sentence. However, we can extract from LNS a level of semantic representation

[4] In LNS, each negative operator corresponds to a single semantic negation; sentences with multiple negative words, but that are semantically negated just once, as in (1), have a single negative operator in LNS. True double negation, as in *we can't not do it* ≈ 'we must do it', requires two negative operators in LNS.

[5] In (21), *nanimo* functions as a negative polarity quantifier modifying an empty head noun, expressed in LNS as _DUMMY; this analysis is motivated by related constructions in Japanese and Chinese.

that expresses only lexical dependencies. This predicate-argument structure (PAS) expresses an important aspect of the semantics of the sentence, viz. who did what to whom, which is useful for applications such as QA. For example, a natural language query of the type "Who shot Lincoln?" is most commonly expressed in Japanese using a cleft construction, as in (23) (*L_Foc* indicates focus):

(23) リンカーンを撃ったのは誰ですか。
rinkaan-o utta-no-wa dare-desu-ka
Lincoln-ACC shot-NML-TOP who-is-QUES

```
FORMULA1 (+Pres +WhQ +Pol L_Foc: NOMINAL1)
|_L_Sub--NOMINAL1 (+Wh)
          |_SemHeads--誰1 (who)
|_SemHeads--FORMULA2 (+Past +Cleft)
          |_L_Sub---_PRO1 (Cntrlr: NOMINAL1)
          |_L_Obj---NOMINAL2
                    |_SemHeads--リンカーン1 (Lincoln)
          |_SemHeads--撃つ1 (shoot)
```

In (23), the information about who did the shooting is only indirectly present: the logical subject of the clause whose semantic head is *utsu* 'shoot' is *_PRO*, an abstract element which must have a Cntrlr; in this case, the Cntrlr of *_PRO1* is NOMINAL1, the focused constituent, whose semantic head is the question word *dare* 'who'.

The PAS derived from (23), shown in (24), yields this information directly, by showing the who-did-what-to-whom information as a set of lexical relations:

(24)
```
撃つ1 (shoot)
|_Dsub----誰1 (who)
|_Dobj----リンカーン1 (Lincoln)
```

All the information expressed by the PAS (24) is inherent in the LNS (23), as PAS is derived from the LNS structure by a language-independent function.

The usefulness of PAS to QA is clear when we consider that the answer to the question in (23) may be found in a database in a variety of forms; the following sentence is a realistic example:

(25) 劇場でリンカーンを撃ったのち、ブースは倉庫に逃げこんだ。
'After shooting Lincoln, Booth ran into a warehouse.'
buusu-wa gekijou-de rinkaan-o utta-nochi,
Booth-TOP theater-at Lincoln-ACC shot-after
souko-ni nigekonda
warehouse-into ran

```
FORMULA1 (+Past +Proposition +I0)
|_L_Sub---_PRO1 (Cntrlr: NOMINAL1)
|_のち(after)+--FORMULA2 (+Past +Proposition +Tme)
          |_L_Sub---NOMINAL1
                    |_SemHeads--ブース1 (Booth)
          |_L_Obj---NOMINAL2
                    |_SemHeads--リンカーン1 (Lincoln)
          |_L_Loc---NOMINAL3
                    |_SemHeads--劇場1 (theater)
          |_SemHeads--撃つ1 (shoot)
|_に(to)+------NOMINAL4
          |_SemHeads--倉庫1 (warehouse)
|_SemHeads--逃げ込む1 (run)
```

The PAS for (25) is shown in (26); the boldfaced part of which matches the query in (24).

(26)
```
逃げ込む1 (run)
|_Dsub----ブース1 (Booth)
|_のち(after) +-----撃つ1 (shoot)
                  |_Dsub----ブース1 (Booth)
                  |_Dobj----リンカーン1 (Lincoln)
                  |_Locn----劇場1 (theater)
|_に+------倉庫1 (warehouse)
```

Extracting only the lexical dependency information in PAS thus contributes to simplifying the interface with applications such as QA.

4.2 Machine translation

The lexical dependency information directly encoded in PAS is not only useful for QA, but to some extent for MT as well. Consider the following example, discussed by Copestake *et al*. (1999): English *white horse* is a translation of German *Schimmel*; in LNS, however, *white* and *horse* are not necessarily in a local relation, since other modifiers may intervene.

(27) a white English horse

```
NOMINAL1 (+Indef +Sing)
|_L_Attrib--FORMULA1
          |_SemHeads--white1
|_SemHeads--NOMINAL2
          |_L_Attrib--FORMULA2
                    |_SemHeads--English1
          |_SemHeads--horse1
```

Copestake *et al*. argue that this shows the need for a flat semantic structure in transfer-based MT, so that there is a direct, local relation between *white* and *horse*. In fact, the PAS derived from (27) provides the flat structure needed:

(28) a white English horse

```
horse1
|_Attrib+-English1
      +-white1
```

In (28) *white* and *horse* are in a local relation, regardless of intervening modifiers.

While such lexical dependency information is certainly useful, it also loses too much information

for the purpose of MT. Consider the English examples in (29) and (30).

(29) It's not him that I like.

```
FORMULA1 (+Pres +Proposition)
|_OpDomain--FORMULA2
            |_L_Foc---NOMINAL1
                        |_SemHeads--he1
            |_SemHeads--FORMULA3(+Pres +Proposition)
                        |_L_Sub---NOMINAL2
                                    |_SemHeads--I1
                        |_L_Obj---NOMINAL3
                                    |_SemHeads--that1
                        |_SemHeads--like1
|_SemHeads--not1 (+F0)
```

(30) It's him that I don't like.

```
FORMULA1 (+Pres +Proposition)
|_L_Foc---NOMINAL1
            |_SemHeads--he1
|_SemHeads--FORMULA2
            |_OpDomain--FORMULA3
                        |_L_Sub---NOMINAL2
                                    |_SemHeads--I1
                        |_L_Obj---NOMINAL3
                                    |_SemHeads--that1
                        |_SemHeads--like1
            |_SemHeads--not1
```

These sentences mean different things, and must be translated differently. An MT system must therefore have access to the scope of negation, information which is available in LNS in (29) and (30), but lost in PAS, which the same for (29) and (30):

(31)
```
like1 (+Neg)
|_Dsub----I1
|_Dobj----he1
```

In this section, we have argued that packaging only the information required by a specific application simplifies the interface between the linguistic component and the application considerably, while still maintaining overall linguistic coherence at the level of LNS. In addition to PAS, other kinds of specialized representations might be extracted from LNS as the need arises from client applications. At this moment, however, the implementation of such representations is left as a future task.

5 Comparison to related work

The distinguishing characteristics of LNS evolved as a result of practical experience: we needed a representational system that is scalable across languages, applications and domains. As noted in Section 1, this combination has led us to a syntactic representation that is abstract enough to be language-neutral, yet shallow enough to allow robust mapping of surface structure to LNS without extensive lexical annotation necessary for deeper semantic analysis. In this section, we compare LNS with similar representational frameworks.

5.1 Semantic representation frameworks

Semantically based representational frameworks include QLF (Alshawi *et al*., 1991; Alshawi and Crouch, 1992), UDRS (Reyle, 1993), Language for Underspecified Discourse representations (Bos, 1995), Minimal Recursion Semantics (Copestake *et al*., 1999) and the Logical Form language of Allen (1995). The distinguishing features of such representations include the use of word-senses as logical predicates, and the explicit logical representation of relations among constituents. LNS differs in both respects: the leaf nodes of an LNS tree are lexemes, not word-senses, and modification relations are not logically characterized.

For example, it is well-known that different kinds of adjectives enter into different semantic relations with the nouns they modify (Keenan and Faltz, 1985), so that the structure of an ADJ+NOUN noun phrase is not sufficient to determine its denotation. Thus *black cat* refers to a cat which is black (i.e., its denotation is given by $\{x \mid \text{black}(x) \wedge \text{cat}(x)\}$), but *legal problem* does not denote $\{x \mid \text{legal}(x) \wedge \text{problem}(x)\}$. To accurately specify the denotations of such NPs would thus require extensive lexical annotation of adjectives, indicating how each adjective sense modifies a noun. Even assuming such fine-grained lexical information at the word sense level, invoking the desirable word sense in the appropriate context would be such a formidable task as to render the system extremely brittle when faced with a realistically broad range of input.

LNS, on the other hand, does not require us to know every kind of relation that an adjective and noun can enter into, since it does not directly provide a truth-functional interpretation. Instead, an adjective modifying a noun is represented as a logical attribute (L_Attrib) modifying a noun or phrase, regardless of the exact semantic relation:

(32) a black cat

```
NOMINAL1 (+Indef)
|_L_Attrib--FORMULA1
            |_SemHeads--black1
|_SemHeads--cat1
```

(33) a legal problem

```
NOMINAL1 (+Indef)
|_L_Attrib--FORMULA1
            |_SemHeads--legal1
|_SemHeads--problem1
```

The fact that LNS is not a semantic representation does not preclude the possibility or desirability of deriving such a representation from LNS; our point is merely that LNS fills a niche as a language-neutral representation without facing the brittleness problem described above.

5.2 Deep syntactic representations

As an abstract syntactic representation, LNS is reminiscent of deep syntactic representations such as f-structure in Lexical Functional Grammar (Bresnan 1982, 2001) and DSyntS (Lavoie and Rambow 1997) based on Dependency Syntax (Mel'čuk 1988). Both these representations try to encode syntactic structure using a language-neutral formal vocabulary, and their benefit to applications such as MT has also been explored (e.g., Han *et al*., 2001). However, LNS differs from these representations in two important respects: (i) Unlike LNS, they have no non-terminal nodes, and hence do not represent the scope of modifiers and operators; in this sense, they are more similar to our PAS representation (Section 4.1). (ii) Surface syntax is more aggressively normalized in LNS than in f-structure or DSyntS: for example, both representations retain the copular *be* and its equivalents, while LNS eliminates them, as they have no semantic function and serve only to satisfy language-specific morphosyntactic requirements (Campbell and Suzuki, 2002). Exactly what is normalized is a matter of degree, yet a higher degree of language-neutrality is generally desirable, not only in principle but also in facilitating multi-lingual applications such as MT.

6 Conclusion and future directions

LNS is a representation framework that is scalable to new languages and to new applications, and robust enough to support broad-coverage systems. Its flexibility and scalability derives from the balance we have tried to strike between a syntactic and a semantic representation. Although semantically motivated, LNS is not a semantic representation per se; and though syntactic, LNS analyses are independent of language-particular grammars.

As currently implemented, LNS is created on a sentence-by-sentence basis: discourse attributes such as Topic are encoded in LNS but not well utilized. In the future we hope to refine LNS so as to better represent aspects of extended discourse, which should enable a more fine-grained analysis of topic/comment structures, anaphora, and so forth. Another area for future development is to incorporate a notion of underspecification in LNS, as developed in many of the semantic frameworks discussed in Section 5.

We see LNS as a representational scheme that can be used by different systems and for different applications. As currently implemented, LNS is also available in XML format, facilitating its portability across systems and applications.

Acknowledgements

We would like to thank the MSR-NLP group, especially Lucy Vanderwende, for comments on earlier conceptions and drafts of the paper. We are also grateful to Mike Calcagno for his help.

References

Allen, J. 1995. *Natural language understanding*. Redwood City, CA: Benjamin/Cummings.

Alshawi, H., D. Carter, M. Rayner and B. Gambäck. 1991. Translation by Quasi Logical Form transfer. In *Proceedings of ACL 29*: 161-168.

Alshawi, H. and Richard Crouch. 1992. Monotonic Semantic Interpretation. *Proceedings of ACL 30*.

Bos, J. 1995. Predicate logic unplugged. In *Proceedings of the 10th Amsterdam Colloquium*, University of Amsterdam.

Bresnan, J. (ed.). 1982. *The Mental Representations of Grammatical Relations*. Cambridge:MIT Press.

Bresnan, J. 2001. *Lexical-Functional Syntax*. Malden, MA and Oxford: Blackwell.

Campbell, R. 2002. Computation of modifier scope in NP by a language-neutral method. COLING 2002.

Campbell, R. and H. Suzuki. 2002. Language-Neutral Syntax: An Overview. MSR Technical Report (in preparation).

Copestake, A., D., Flickinger, I. Sag and C. Pollard. 1999. Miminal Recursion Semantics: An Introduction. Ms., Stanford University.

Keenan, E.L. and L.M. Faltz. 1985. *Boolean semantics for natural language*. Dordrecht: D. Reidel.

Han, C-H., B. Lavoie, M. Palmer, O. Rambow, R. Kittredge, T. Korelsky, N.Kim and M. Kim. 2001. Handling Structural Divergences and Recovering Dropped Arguments in a Korean/English Machine Translation System. In *Proceedings of AMTA*.

Heidorn, G. 2000. Intelligent Writing Assistance. In R. Dale, H. Moisl and H. Somers (eds.), *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, New York: Marcel Dekker.

Lavoie, B. and O. Rambow. 1997. A fast and portable realizer for text generating systems. In *Proceedings of ANLP'97*.

Mel'čuk, I. 1988. *Dependency Syntax: Theory and Practice*. New York: State University of New York Press.

Reyle, U. 1993. Dealing with ambiguities by underspecification: construction, representation and deduction. *Journal of Semantics* 10: 123-179.

Shibatani, M. 1990. *The Languages of Japan*. Cambridge: Cambridge University Press.