RC24046 (W0609-009) September 6, 2006
Computer Science

IBM Research Report

Implicit Predicate Arguments and Discourse

Arendse Bernth
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

LIMITED DISTRIBUTION NOTICE: This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties). Copies may be requested from IBM T. J. Watson Research Center , P. O. Box 218, Yorktown Heights, NY 10598 USA (email: reports@us.ibm.com). Some reports are available on the internet at http://domino.watson.ibm.com/library/CyberDig.nsf/home .

Implicit Predicate Arguments and Discourse

Arendse Bernth*

* *IBM T.J. Watson Research Center*
19 Skyline Dr., Hawthorne, NY 10532
USA

arendse@us.ibm.com

ABSTRACT. *An important aspect of semantic analysis of a discourse is the identification and resolution of implicit arguments of verbs, nouns etc. Making implicit arguments explicit enables extraction of relations that would otherwise be hidden to an application of the semantic analysis such as question-answering. We describe how the discourse understanding system Euphoria uses parsing, syntactic rules, semantic rules, and coreference resolution to determine implicit arguments. The semantic representation for a discourse consists of a set of* entity-oriented logical forms *indexed by the entities of the discourse, enabling efficient access to everything that is said about each discourse entity. Empirical evaluation of Euphoria shows significant improvements in resolving implicit arguments as compared to sentence-based processing.*

KEYWORDS: *discourse representation, implicit arguments, logical form, semantic analysis.*

1. Introduction

An important aspect of semantic analysis of a discourse is the identification and resolution of implicit arguments of verbs, nouns etc. Making implicit arguments explicit enables extraction of relations that would otherwise be hidden to an application of the semantic analysis, such as question-answering.

Discourse structure requires a semantic representation of a complete discourse, showing not only the overall structure of the discourse but also the logical representation of the individual sentences. Often, arriving at the correct semantic representation of individual sentences, including making implicit arguments explicit, requires analysis beyond individual sentences, looking at discourse-global phenomena.

In this paper we describe how the computational discourse understanding system Euphoria (Bernth, 2002; Bernth, 2004) uses both local and discourse-level analysis to identify and resolve implicit arguments.

Euphoria is a computational discourse understanding system that takes as input a discourse of several sentences, typically the length of a medical abstract or newspaper article, and produces a semantic analysis for the whole document. It is built on top of the English Slot Grammar (ESG) (McCord, 1980; McCord, 1990; McCord, 1993; McCord, 2006a; McCord, 2006b; McCord, 2006c).

Slot Grammar is a dependency-oriented grammatical system, where analysis is driven by *slots* associated with head words. A word (sense) can have two kinds of slots–*complement* slots, associated with the word sense in the lexicon, and *adjunct* slots, associated with the part of speech of the word sense in the grammar. Complement slots have a dual role–as grammatical relations and as names for logical arguments of a word sense. Slot Grammar parsing is normally done with a bottom-up chart parser, where the basic method for combining two phrases is to let one fill a slot of the other one.

Whereas ESG delivers a *syntactic* analysis on a *sentence* level, Euphoria produces a *semantic* analysis spanning *several* sentences with coreference resolved and implicit arguments made explicit. The semantic interpretation is based on the ESG parses, but utilizes most-plausible semantics to override the parses in some cases. During processing, Euphoria also makes use of discourse constraints, selectional constraints, and corpus-based statistics. For coreference, an enhanced version of the system described in Bernth (2002) is used.

After giving a general introduction to the semantic analysis produced by Euphoria, this paper focuses on the semantic analysis for implicit arguments of verbs, adverbs and adjectives, and nouns.

Euphoria's semantic analysis is expressed in terms of *entity-oriented logical forms* (*EOLF*s), which make use of *extended entities* (*EE*s). EEs include not only entities in the conventional sense (including named entities), but also events and relations. They are basically anything that can be referred to. One of the major foci of Situation Semantics (Barwise *et al.*, 1983) was that most classes of words are referential, a point we agree with. The consequence of this view is a need to make all types of entities referrable. The solution in Situation Semantics was to utilize the notion of realism, a computationally somewhat vague idea. The same objective can be accomplished through the idea of indexing, first proposed by Davidson (1967). Davidson's original idea covered indexing of verbs by so-called event variables, an idea that can be generalized to other entity types. Our "events" are indeed very general, along the lines described in Hobbs (1985) and McCord *et al.* (2005).

This notion of such generalized "events" furthermore has the advantage of allowing a "flat" semantic structure, a property that makes automatic reasoning easier. Flat structures are also used in Hobbs (1985), but there are differences. Both Situation Semantics and Discourse Representation Theory (DRT) (Kamp, 1981) point out the necessity of interpreting a discourse in context. Like DRT, Euphoria builds up the discourse structure within the context of the preceding discourse, but also takes some later discourse into consideration for certain types of cataphora. Other divergences from a DRT-like representation include the generalization of events as mentioned above, and the use of a type-free semantic representation.[1]

In order to identify and resolve the implicit arguments, Euphoria employs ESG parsing, syntactic and semantic rules, and coreference resolution. ESG parsing is used in two ways. First, ESG actually explicitly provides some implicit arguments. Second, for the cases where ESG does *not* identify or provide implicit arguments, the parse is exploited by syntactic rules in Euphoria to identify the implicit arguments. However, it is not enough to *identify* the–sometimes ellipted–words in the sentence that constitute the implicit arguments. It is also necessary to *resolve* the coreference of these words with previous words in the document to ensure that the correct discourse entities are used in the semantic analysis.

In this paper, the focus is on finding the syntactic constituents that represent the implicit arguments and on applying semantic constraints in case disambiguation is needed. Euphoria's coreference module is then called upon to supply the information about the correct discourse entity for that constituent. The interested reader is referred to Bernth (2002) for a description of the coreference aspect.

The examples of actual EOLFs in this paper are all produced by Euphoria. In order to maximize readability, most of the examples are of the minimal size necessary to illustrate the point in question.

Section 2 describes the entity-oriented logical forms and their components. In section 3 the treatment of implicit arguments is described, and our results are stated and discussed in section 4. Section 5 gives our conclusion.

---

1. See Menzel (1986) for the advantages of a type-free semantic representation for natural language.

2. Entity-oriented logical forms

In this section we describe the entity-oriented logical forms and their components.

An *entity-oriented logical form* (EOLF) consists of an extended entity E (called the *index* of the EOLF), together with a set *S* of *predications* that are "about" E, in the sense that E appears in each member of *S*.[2]

Each predication in an EOLF is of the form

(1) $(e\ arg_1\ arg_2\ \ldots\ arg_n)$

where both the predicate e and the arguments $arg_1, \ldots, arg_n$ are extended entities (EEs) (*n* may be 0). For example, *e* could be an event of seeing in the usual sense of *see*, with arg_1 as the subject, arg_2 as the object, and *n* = 2. Arguments generally follow the order of ESG slot filler arguments.

So we are using the main entity like the event arguments of Davidson (except that we encode the word sense predicate with the entity argument), but they are not restricted to events. They can name/index any entity in the ontology, where **entity** is the unique top node of the ontology.

The semantic analysis of a document (or collection of documents) produced by Euphoria is a list of EOLFs that express the semantic content of the document. The list is *entity-oriented* in two senses:

– Every entity E mentioned in the document has associated with it an EOLF for which it is the index: All the things that are said about E are listed with E and are accessible efficiently from E.

– For the predicates that appear in the EOLFs, both the predicates and their arguments are entities, except for certain special predicates described in section 2.3.

Entity types are described in section 2.1, and special entities and special predicates are listed in section 2.2 and section 2.3, respectively.

2.1. *Entity types*

In this section we describe the various entity types. Even though no type conformance is required for entities, it is useful to retain a trace of the *kind* of entity, as indicated by the part of speech of the word that gives rise to the entity.[3] This is useful for both text generation from the EOLF and for constraining inference.

Generally, entities are given reference identifications (*refID*s, for short) of the form *word#NT*, e.g. *see#33V*, where *word* is derived from the first mention in the text of

2. In the examples, we will display the EOLFs as follows: `Index` $<$ `(Predication`$_1$ `...` `Predication`$_n$`)`.
3. Deverbal nouns are represented as verbs, as we shall see later in this paper.

the entity, *N* is a unique number, and *T* is one or more characters indicating the part of speech of the first mention.

The following types are currently produced:

A Adjective. Example: *small#2A*. Adjectives and adverbs may furthermore be marked *C* or *S* for *comparative* and *superlative*, respectively.

Adv Adverb. Example: *quickly#5Adv*.

G Generic. Example: *elephant#1G*.

P Preposition. Example: *above#15P*.

V Verb. Example: *see#3V*. Verbs occurring within intensional contexts, for instance *reads* in *John believes that Mary reads*, may further be marked with *i* as in *read#33Vi*. Currently only the head of the embedded clause is marked with *i*, and from this it is possible for the user to infer that all the embedded entities are within the intensional context. In future versions of Euphoria we may mark *all* embedded entities, for the convenience of the user. We do not currently have any plans to disambiguate the intensional context.

If no type is given, the type defaults to noun. Example: *house#120*. Note that currently generics are assumed to be derived from nouns only.

2.2. *Special entities*

In addition to the entities directly reflecting mentions in the text, there are the following special entities:

u An unfilled argument. Example: *(eat#5 u mango#4)*. Here the first argument, the deep subject, is unspecified. This could stem from a passive construction like *The mango was eaten*.

you-imp The implicit subject of an imperative verb. For example, *Read the book!* is represented as *(read#2V you-imp book#1)*. See section 3.1.2.

year#n Entity for time expressions involving a year. *n* is (a string representing) an integer indicating the year. For example, *year#1998* means the year *1998*.

month#n Entity for time expressions involving a month. *n* is an integer indicating the month. For example, *month#12* means the month *December*.

date#N Entity for general date expression. *N* is a term indicating the date. Following ESG conventions about naming of dates, dates are given as *(WD MD M Y)* where *WD* is weekday, *MD* is day of the month, *M* is month, and *Y* is year. Values are integers. *WD* starts with 1 for Monday and ends with 7 for Sunday;

month and year as above. If a field is not available it is set to 0 (zero). An example of a fully resolved date is *date#(4 12 8 2004)*, which means *Thursday, August 12, 2004*. This entity is used for recording the results of resolving time expressions, including the expressions *today*, *yesterday* and *tomorrow*.

2.3. *Special predicates*

Entities, which may function as predicates, are derived from specific mentions in the text. However, there are also a number of special predicates, which are not entities, and which do not *directly* reflect any mentions, but rather derived attributes. The most important of these are:

card This predicate indicates the *cardinality* of an entity. We will view entity arguments as sets of individuals where an entity that is an individual is represented as a singleton set. For example, *(card pilot#2 sing)* states that the cardinality of the entity *pilot#2* is singular. Other values for *card* may be *plur*, a specific number, or a generalized quantifier such as *many*. *Negation* is considered a special case where *card* is zero; hence the cardinality of the seeing event *see#3V* in example (2) is zero. For verbs, the cardinality is only given in case of negation, and not for positive statements. Note also that for negation, the cardinality is always attached to the verb, and this leaves undecided what the scope of the negation is.

```
(2) a. John did not see Mary.
    b. John#1 < ((see#3V John#1 Mary#2 u)(card John#1 sing))
       Mary#2 < ((see#3V John#1 Mary#2 u)(card Mary#2 sing))
       see#3V < ((see#3V John#1 Mary#2 u) (card see#3V 0))
```

dest This predicate specifies the *destination*. For example *(dest representative#2 send#3V)* in example (3) means that *representative#2* is the destination of the sending event *send#3V*.

```
(3) a. The letter was sent to the representative.
    b. letter#1          < ((send#3V u letter#1 u)
                            (card letter#1 sing))
       representative#2 < ( (dest representative#2 send#3V)
                            (card representative#2 sing))
       send#3V           < ((send#3V u letter#1 u)
                            (dest representative#2 send#3V))
```

instr This predicate gives the *instrument*. For example, *(instr eat#4V fork#11)* means that the instrument of *eat#4V* is *fork#11*.

is_in This predicate specifies a geographical location within another geographical location. For example, *(is_in Morocco#9 Marrakech#8)* means that *Marrakech#8* is in *Morocco#9*.

loc This predicate indicates the *location*. For example, *(loc Alaska#1 snow#4V)* means that the location of the snowing event *snow#4V* is *Alaska#1*.

poss This predicate indicates *possession*. It may reflect an *'s*-possessive in the text, a possessive pronoun, or the verb *have* with an object. For example, *John has a house*, *John's house*, and *His house* (assuming that *His* is coreferential with *John*)[4] will all produce the predication *(poss John#1 house#2 u)*.

time This predicate indicates the *time*. For example, *(time year#1991 cross#19V)* means that the time of the entity *cross#19V* is *1991*.

3. Implicit predicate arguments

As part of the disambiguation of the text, Euphoria identifies and resolves a number of predicate arguments that are implicit in the surface structure. Some of these arguments are derived directly from the deep level of the ESG parses; others are decided on by Euphoria.

Words that may have implicit arguments have in common that they take complement slots. The open-class words–verbs, nouns, adjectives, adverbs–can all have complement slots and implicit arguments. Implicit subjects and objects of verbs are described in section 3.1. Our treatment of implicit arguments for adverbs and adjectives appears in section 3.2, and section 3.3 describes noun arguments.

3.1. *Verbs*

Implicit arguments for verbs occur with the nonfinite forms and the imperative mood. We shall consider here implicit subjects of infinitives, imperatives, and present participles, as well as implicit deep objects of passive past participles, which appear as subjects on the surface level.

The rest of this section is organized as follows. Section 3.1.1 describes infinitival verb complements; section 3.1.2 implicit subject of imperatives; section 3.1.3 describes present participles; and section 3.1.4 the implicit deep object of past participles.

3.1.1. *Infinitives*

Maybe the simplest example of an implicit subject for an infinitive, syntactically speaking, is a sentence like *John can swim*, where the bare infinitive *swim* is a com-

4. Also assuming that this is an attributive statement (Donnellan, 1966).

plement of the modal verb *can*. Consider the more complicated example in (4), where the main verb has both an object, *Mary*, and an infinitive complement *swim*. Since *swim* is a complement of *see*, it shows up in the ESG argument list of that word. And *Mary* is identified as the implicit subject of *swim*.[5]

```
(4) a. John sees Mary swim.
    b. John#1  < ((see#3V John#1 Mary#2 swim#4V))
       Mary#2  < ((see#3V John#1 Mary#2 swim#4V)
                  (swim#4V Mary#2 u))
       see#3V  < ((see#3V John#1 Mary#2 swim#4V))
       swim#4V < ((see#3V John#1 Mary#2 swim#4V)
                  (swim#4V Mary#2 u))
```

Infinitives can also be complements of verbs which require the infinitive marker *to*. Example (5) shows a simple example of this. Here *John* is identified as the subject of *swim*. And since *want* creates an intensional context, the refID for *swim* is marked with an *i*, as described in section 2.1.

```
(5) a. John wants to swim.
    b. John#1   < ((want#2V John#1 swim#3Vi)
                   (swim#3Vi John#1 u))
       swim#3Vi < ((want#2V John#1 swim#3Vi)
                   (swim#3Vi John#1 u))
       want#2V  < ((want#2V John#1 swim#3Vi))
```

Example (6) combines the cases illustrated in example (4) and example (5). The main verb has both an object and an infinitive complement, and the context is intensional. *Mary* is identified as the implicit subject of *swim*.

```
(6) a. John wants Mary to swim.
    b. John#1   < ((want#3V John#1 Mary#2 swim#4Vi))
       Mary#2   < ((want#3V John#1 Mary#2 swim#4Vi)
                   (swim#4Vi Mary#2 u))
       swim#4Vi < ((want#3V John#1 Mary#2 swim#4Vi)
                   (swim#4Vi Mary#2 u))
       want#3V  < ((want#3V John#1 Mary#2 swim#4Vi))
```

The implicit subjects in the above examples are all identified by ESG, based on lexical and syntactic information.

Infinitival complements of adjectives pose special problems. Example (7) illustrates the simplest case where the implicit subject of *go* clearly is *John*.

5. We shall not go further into the semantics of perception verbs in this paper.

```
(7) a. John was clever to go.
    b. John#2    < ((clever#1A John#2 go#3V) (go#3V John#2 u))
       clever#1A < ((clever#1A John#2 go#3V))
       go#3V     < ((clever#1A John#2 go#3V) (go#3V John#2 u))
```

In example (7) we can justifiably conclude that John actually goes. However, this is by no means *always* the case. Some adjectives clearly do not allow us to conclude that the activity described by the infinitive complement actually took place. If we substitute *afraid* for *clever* in example (7) we get *John was afraid to go*. Depending on context, John may or may not have gone. Additionally, according to our scheme of marking verbs within intensional contexts with an *i*, we should really mark the refID for *go* with an *i* in this case.

In the above examples involving complements of adjectives the overt subject of the sentence provided the implicit subject of the infinitive. This may not always be the case. Consider the sentence *The book was enjoyable to read*. Here the overt subject *the book* supplies not the implicit subject of *read*, but rather the object. The distinction appears to be *semantic*, and we shall see in section 3.1.3 how a lexicon of *selectional preferences* can assist in making a decision in semantic cases.

3.1.2. *Imperatives*

Whereas ESG provides the implicit arguments of some of the infinitival complements (without the coreference aspect) described above in section 3.1.1, it leaves the implicit subject of imperatives unspecified. This implicit subject is the addressee of the utterance, and we assign a special entity *you-imp* to designate such an implicit subject, as illustrated by example (8). Furthermore we will assume that this implicit subject is of semantic type *human* and use this semantic type for any applications of the selectional constraints lexicon.

```
(8) a. Inflate the balloon!
     b. balloon#1  < ((inflate#2V you-imp balloon#1)
                       (card balloon#1 sing))
        inflate#2V < ((inflate#2V you-imp balloon#1))
```

3.1.3. *Present participles*

Present participles following an object pose an interesting ambiguity in that they can be attached either to the subject or the object. The implicit subject of the participle depends on the attachment. The controlled-language checker EEA (Bernth, 1997) identifies this type of ambiguity, and offers disambiguated rewriting suggestions reflecting the different attachment possibilities, but does not make a decision on which attachment is correct. Using techniques similar to those described in Bernth (1998) for EEA to *identify* the ambiguity, and the lexicon of selectional preferences described in Bernth *et al.* (2003) to *make a choice*, we can resolve the implicit subject. The lexicon provides us with class-based preferences for the semantic types of the complements of a verb. For example, the verb *eat* strongly prefers an object of type *food* and a subject of type *animate*, even though there are exceptions, of course.

An example of resolving an implicit subject of a present participle is given in (9).

This sentence is structurally ambiguous. Who is *wearing old shoes and rubber gloves*? There are two possibilities here: either the subject in the main clause, which is the implicit subject of *Harvest*, or the object, which is *walnuts*. We note that Euphoria has chosen the implicit subject of *Harvest* as shown by the first argument of *wear#7V* being *you-imp*.

```
(9) a. Harvest the walnuts wearing old shoes and rubber gloves.
    b. and#4            < ((wear#7V you-imp and#4 u)
                            (and#4 shoe#3G rubber gloves#5G))
       harvest#6V       < ((harvest#6V you-imp walnut#2 u))
       old#1A           < ((old#1A shoe#3G))
       rubber gloves#5G < ((and#4 shoe#3G rubber gloves#5G))
       shoe#3G          < ((old#1A shoe#3G)
                             (and#4 shoe#3G rubber gloves#5G))
       walnut#2         < ((harvest#6V you-imp walnut#2 u)
                             (card walnut#2 plur))
       wear#7V          < ( (wear#7V you-imp and#4 u))
```

This example illustrates how Euphoria in certain cases overrides the ESG parse. According to the common parsing heuristic of preferring close attachment, the *walnuts* are accoutred with shoes and gloves. Although ESG uses several heuristics for attachment, the one that applies in this case is close attachment, and the ESG parse is shown in (10).[6] However, real-world knowledge tells us that humans are much more likely to wear gloves than walnuts are.

(10) "Harvest the walnuts wearing old shoes and rubber gloves."

```
o--------- top    harvest1(1,u,3) verb vimpr human_agent nhuman_object
|                                 (harvest#6V)
| .------- ndet   the1(2)         det pl def the ingdet
'-+------- obj(n) walnut1(3)      noun cn pl st_tree st_nut (walnut#2)
  '------- nnfvp  wear1(4,3,7,u)  verb ving (wear#7V)
   |   .- nadj    old1(5)         adj erest adjnoun (old#1A)
   | .--- lconj   shoe1(6)        noun cn pl st_shoe (shoe#3G)
   '-+--- obj(n)  and0(7)         noun cn pl cord st_shoe st_clothes
     |                            (and#4)
     | .- nnoun   rubber1(8)      noun cn sg massn
     '--- rconj   glove1(9)       noun cn pl st_clothes (rubber
                                  gloves#5G)
```

6. See appendix A for an explanation of how to read ESG parse trees.

The confidence score given by the selectional constraints lexion for humans wearing gloves is 0.101124 whereas the confidence score for walnuts is only 0.001873; hence the attachment of *wearing* to the implicit subject of *Harvest* is preferred, and the implicit subject of *wearing* is determined to be *you-imp*. The implicit subject of the present participle *flying* in example (11) is also resolved using this technique.

A similiar technique is applied to prepositional phrase attachment.

3.1.4. *Past participles*

Present participles are inherently active in nature. Let us now look at passive past participles, as exhibited in in example (11). Here the deep subject position for *dip* is unfilled, as indicated by the presence of *u*, whereas the deep object position is filled by *wing#9*. Furthermore we note that the implicit subject of *flying* is correctly resolved; this is a case of Euphoria using most-plausible semantics (selectional preferences) to override the ESG parse. Also note that Euphoria correctly resolves the referent of *its* to be the plane rather than the mountain.

```
(11) a. The plane hit the mountain flying with its right wing
   dipped downwards.
     b. dip#7V         < ((dip#7V u wing#9 u)
                          (downwards#1Adv dip#7V))
        downwards#1Adv < ((downwards#1Adv dip#7V))
        fly#6V         < ( (fly#6V plane#3 u u)
                          (with#8P wing#9 fly#6V))
        hit#5V         < ((hit#5V plane#3 mountain#4 u))
        mountain#4     < ((hit#5V plane#3 mountain#4 u)
                          (card mountain#4 sing))
        plane#3        < ((hit#5V plane#3 mountain#4 u)
                          (card plane#3 sing)
                          (fly#6V plane#3 u u)
                          (poss plane#3 wing#9))
        right#2A       < ((right#2A wing#9))
        wing#9         < ((poss plane#3 wing#9)
                          (with#8P wing#9 fly#6V)
                          (right#2A wing#9) (dip#7V u wing#9 u)
                          (card wing#9 sing))
        with#8P        < ((with#8P wing#9 fly#6V))
```

3.2. *Adverbs and adjectives*

Comparative and superlative adverbs and adjectives also raise some interesting issues. Consider the example in (12) involving the adverb *faster*. This is a kind of elliptical construction in that there actually are two *progress* events, one involving *Lisa* as the subject, and one involving *anyone*.[7] Given a construction like this, Euphoria will introduce a new refID for the ellipted event and try to resolve any implicit arguments by exploring the parse tree. In (12) Euphoria introduces the new refID *progress#8V*, which is then used in the predication for the comparative adverb: *(faster#6AdvC progress#4V progress#8V).*

```
(12) a. Lisa has progressed faster than anyone Nowak ever
        mentored.
     b. Lisa#1        < ((progress#4V Lisa#1) (card Lisa#1 sing))
        Nowak#3       < ((mentor#5V Nowak#3 anyone#2)
                         (card Nowak#3 sing))
        anyone#2      < ((progress#8V anyone#2)
                         (card anyone#2 sing)
                         (mentor#5V Nowak#3 anyone#2))
        ever#7Adv     < ((ever#7Adv mentor#5V u))
        faster#6AdvC  < ((faster#6AdvC progress#4V progress#8V))
        mentor#5V     < ((ever#7Adv mentor#5V)
                         (mentor#5V Nowak#3 anyone#2))
        progress#4V   < ((progress#4V Lisa#1)
                         (faster#6AdvC progress#4V progress#8V))
        progress#8V   < ((progress#8V anyone#2)
                         (faster#6AdvC progress#4V progress#8V))
```

A similar construction occurs with comparative adjectives, as in *Lisa's progress was faster than for anyone Nowak ever mentored.*

3.3. *Nouns*

Also nouns can have implicit arguments. Like verbs, these nouns indicate *relations*, and the implicit arguments are what are being related. In many instances nouns with implicit arguments do have a verb counterpart. For example, a noun such as *activation* may be regarded as a variation of the verb *activate*.[8] The noun has an implicit argument of something that is being activated and one of something that does that activation. Not only will we term these implicit arguments (deep) "object" and (deep) "subject" respectively, but also make an actual conversion of part of speech from noun to verb for the refID.

There are several steps involved in our treatment of deverbal nouns:

– Recognize the proper conditions under which a noun has implicit arguments.

7. Note that Euphoria does not yet do much with bound anaphora such as *anyone* in this example.
8. We shall loosely use the term "deverbal noun" without committing to whether the noun or the verb seems more basic.

– Identify the implicit arguments.
– Convert the noun to a verb and construct the proper predication.

Conditions for implicit arguments As mentioned earlier in this section, a deciding factor is that the noun can take complement slots. For our purposes we shall consider only the case of one complement slot, the `nobj` slot, and when this is actually filled. Most commonly, this slot will be filled by an *of*-PP, but other prepositions such as *to* as in *reference to the book* are also possible. Additionally, the noun should be convertible to a verb, as described below.

Identification of implicit arguments In tandem with the identification of the `nobj` slot we can also explore the parse tree to find the deep object which–for cases of objective genitives–is the prepositional object of the `nobj`-phrase.[9] The deep subject is often the subject of the verb phrase that the noun occurs in, if there is such a verb phrase. Otherwise it could be anaphoric or unfilled.

For cases of subjective genitive, the prepositional object is the subject, and the mother of the `nobj` the object. Objective genitives are more common, and for transitive verbs, the default is to use the objective interpretation. However, intransitive verbs *must* receive a subjective genitive interpretation. Additionally, sentences like *The company's target of 10 percent subscriber growth is achievable* require a subjective genitive interpretation with the deverbal noun being *target*, the subject being the *'s*-genitive, and the object *growth*.

It is also important to identify partitive constructions such as *half of the growth*. Even though *half* has a valid verb correspondence, *halve*, the sense used with *of*-constructions is not the sense that one would want to convert into a verb.

Conversion to verb This is done by lexical lookup in a dictionary of nouns and their corresponding verbs derived from WordNet (Fellbaum, 1998). If this fails, a number of morphological rules are applied. Verb candidates generated by the rules are validated by lexical lookup in the base ESG dictionary.

Let us look at the example given in (13), together with its parse produced by ESG.

The two nouns *activation* and *dissociation* have implicit arguments that can be derived from the parse, and the refIDs marked on the tree for *activation* and *dissociation* show the conversion to verbs by Euphoria. Furthermore, it is worth noting that ESG supplies the implicit argument of *modifying* (the subject).

A subset of the EOLFs produced by Euphoria is given in (14).[10] Here we note that the implicit arguments of *activation* and *dissociation* have been identified and filled in. This is done by going through the steps of exploring the parse to find the `nobj`

9. As argued by Quirk *et al.* (1972), there is a very close correspondence between a number of *of*-constructions and genitive, so we take the liberty of applying the term "genitive".
10. Some parts that are irrelevant for illustrating the handling of implicit arguments have been removed for readability.

complement for each of the two nouns and deciding that *activation* and *dissociation* are deverbal nouns, as described above.

(13) “An active phorbol ester must therefore presumably by activation of protein kinase cause dissociation of a cytoplasmic complex of NF-kappa B and L kappa B by modifying L kappa B.”

```
  .--------------- ndet          an1(1)             det sg indef
  .--------------- nadj          active1(2,u)       adj (active#1A)
  .--------------- nadj          phorbol(3)         noun propn sg (b#8)
.----------------- subj(n)       ester1(4)          noun cn sg (ester#3)
o----------------- top           must1(5,4,13)      verb vfin vpres sg
`----------------- vadv          therefore1(6)      adv (therefore#13Adv)
`----------------- vadv          presumably1(7)     adv
|                                                   (presumably#14Adv)
`----------------- vprep         by1(8,9)           prep pprefv
| `--------------- objprep(n)    activation1(9,10)  noun cn sg (activate#4V)
|   `------------- nobj(n)       of1(10,12)         prep pprefn nonlocp
|     | .--------- nnoun         protein1(11)       noun cn sg
|     `----------- objprep(n)    kinase1(12)        noun cn sg
|                                                   (protein kinase#5)
`----------------- auxcomp(binf) cause1(13,4,14,u)  verb vinf (cause#11V)
  `--------------- obj(n)        dissociation1(14)  noun cn sg
  | |                                               (dissociate#6V)
  | `------------- nobj(n)       of1(15,18)         prep pprefn nonlocp
  |   | .--------- ndet          a1(16)             det sg indef
  |   | .--------- nadj          cytoplasmic1(17)   adj (cytoplasmic#2A)
  |   `-+--------- objprep(n)    complex2(18,19)    noun cn sg (complex#7)
  |     `--------- nobj(n)       of1(19,23)         prep pprefn nonlocp
  |       |     .- nadj          NF(20)             noun propn sg notfnd
  |       |   .--- nadj          kappa1(21)         noun cn sg
  |       | .----- lconj         b1(22)             noun cn sg (b#8)
  |       `-+----- objprep(n)    and0(23)           noun cn pl (and#9)
  |         | .--- nadj          L1(24)             noun propn sg
  |         | .--- nnoun         kappa1(25)         noun cn sg
  |         `----- rconj         b1(26)             noun cn sg
  |                                                 (kappa B#10)
  `--------------- vprep         by1(27,28)         prep pprefv
    `------------- objprep(ing)  modify1(28,4,30)   verb ving
      |                                             (modify#12V)
      | .--------- nadj          L1(29)             noun propn sg
      `-+--------- obj(n)        kappa1(30)         noun cn sg
        |                                           (kappa B#10)
        `--------- nprop         B1(31)             noun propn sg
                                                    (kappa B#10)
```

```
(14) a. An active phorbol ester must therefore presumably by
        activation of protein kinase cause dissociation of a
        cytoplasmic complex of NF-kappa B and L kappa B by
        modifying L kappa B.

     b. activate#4V    < ((instr activate#4V cause#11V)
                          (activate#4V ester#3 protein kinase#5G))
        cause#11V     < ((instr activate#4V cause#11V)
                          (cause#11V ester#3 dissociate#6V)
                          (instr modify#12V cause#11V))
        dissociate#6V < ((cause#11V ester#3 dissociate#6V)
                          (dissociate#6V ester#3 complex#7))
        ester#3       < ((active#1A ester#3)
                          (activate#4V ester#3 protein kinase#5)
                          (card ester#3 sing)
                          (cause#11V ester#3 dissociate#6V)
                          (dissociate#6V ester#3 complex#7)
                          (modify#12V ester#3 kappa B#10))
        kappa B#10    < ((and#9 b#8 kappa B#10)
                          (modify#12V ester#3 kappa B#10)
                          (card kappa B#10 sing)
                          (kappa#15 u kappa B#10))
        modify#12V    < ((instr modify#12V cause#11V)
                          (modify#12V ester#3 kappa B#10))
```

4. Results

This section describes the results of evaluating Euphoria. The results are of two kinds: the numerical results described in section 4.1 and conclusions about promising areas for future work described in 4.2.

4.1. *Evaluation of determination of implicit arguments*

One of the important measures of a system's performance is precision, defined as the proportion of "answers" given by the system that are correct. Another important measure is recall, defined as the proportion of the correct "answers" found, relative to the total number of answers present in the document.

Obviously there is a trade-off between recall and precision; hence the *F-measure*, which provides a weighted measure combining both precision and recall, is also of interest. In our results we give the weighted harmonic mean of precision and recall, an F-measure where precision and recall have equal weight.

For purposes of this paper, an “answer” is identification and resolution of an implicit argument. This means that we are measuring how many of the implicit arguments returned by Euphoria are resolved correctly in the context of the whole discourse (precision) and how big a percentage of the actual implicit arguments in the documents are correctly resolved (recall).

Given the nature of the task, we opted for hand evaluation of a limited number of documents. Euphoria was trained on a variety of documents, mostly news articles and medical abstracts. For the evaluation, we picked 10 unseen news articles from the English Gigaword corpus with a total of 325 sentences.

The results are given in table 1.

	Precision	**Recall**	**F-measure**
ESG Baseline	0.4161	0.4685	0.4407
Euphoria	0.6062	0.9212	0.7312

Table 1. *Evaluation results.*

For our baseline, we measured the performance of ESG alone, but on the document level. That is, referents have to be correct, and arguments correctly identified and resolved. This reflects the task at hand; ESG does not attempt to resolve reference. Referring to table 1, we see that the results for this were precision 0.4161 and recall 0.4685.

Referring to this baseline and the results for Euphoria given in table 1, we see that adding Euphoria raised the precision from 0.4161 to 0.6062 and the recall from 0.4685 to 0.9212, a significant improvement.

4.2. *Future work*

The evaluation and error analysis identified a number of areas that can be fruitfully addressed in order to improve the system. In this section we describe the most important ones.

The coverage of syntactic rules in Euphoria needs to be expanded to handle more constructions. These rules should particularly be targeted towards implicit noun arguments.

Resolving the internal structure of noun phrases involving compound nouns is beyond the scope of this paper. ESG does not provide much support for this. A first approximation would be for Euphoria to include an objective genitive interpretation of $Noun_1$ $Noun_2$ constructions where the head noun ($Noun_2$) is deverbal and takes an `nobj`-complement.

Resolving some cases requires more semantic analysis. An example is the resolution of the arguments of *eat* in (15) and (16).

(15) "John was ready to eat."

(16) "The bread was ready to eat."

In (15) *John* is the subject of *eat*, whereas in (16), *bread*, in a similar surface role as *John*, is the object. Presumably the lexicon of selectional preferences can assist with cases like this. More difficult cases like *The chicken is ready to eat* require much deeper analysis of the context in order to decide whether the chicken is the subject or object.

The error analysis revealed that a mismatch of word senses was a source of problems. For example, in associating verbs with deverbal nouns, one really needs to look at the *sense* of a noun to know whether it is deverbal. Euphoria does not disambiguate word senses, so we could not use such information, and used instead a simpler noun-verb correspondence dictionary that did not take advantage of the full word sense information that could be extracted from WordNet. As a result of this, some nouns were erroneously determined to be deverbal, even though that was clearly not the case in context. Some examples are *amount*, *middle* and *total*. Disambiguating word senses would also contribute to the general goal of Euphoria of providing a disambiguated semantic analysis, which would not be complete without disambiguation on the lexical level as well as on the structural and referential levels.

To summarize, the identified areas of work that should improve the system are as follows:

– Expand coverage of Euphoria's syntactic rules.
– Improve the coreference.
– Increase use of the lexicon of selectional preferences.
– Add word sense disambiguation.

These provide promising areas for future research.

5. Conclusion

We have shown how taking into account the whole discourse significantly improves the resolution of implicit arguments in predicate argument structure. This is important because making implicit arguments explicit provides substantially more information for any application of a discourse understanding system.

6. References

Barwise J., Perry J., *Situations and Attitudes*, MIT Press, New York, NY, 1983.

Bernth A., " EasyEnglish: A Tool for Improving Document Quality", *Fifth Conference on Applied Natural Language Processing*, Association for Computational Linguistics, Washington, DC, USA, p. 159-165, 1997.

Bernth A., " EasyEnglish: Addressing Structural Ambiguity", *in* D. Farwell, L. Gerber, E. Hovy (eds), *Machine Translation and the Information Soup, Third Conference of the Association for Machine Translation in the Americas*, n˚ 1529 in *Lecture Notes in Artificial Intelligence*, Association for Machine Translation in the Americas, Springer, Langhorne, PA, USA, p. 164-173, 1998.

Bernth A., Euphoria – A Reference Resolution System for Machine Translation, Technical Report n˚ RC22627, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, November, 2002.

Bernth A., Euphoria Semantic Analysis, Technical Report n˚ RC23396, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, November, 2004.

Bernth A., McCord M. C., " A Hybrid Approach to Deriving Selectional Preferences", *Proceedings of MT Summit IX*, New Orleans, p. 9-15, 2003.

Davidson D., " The Logical Form of Action Sentences", *in* N. Rescher (ed.), *The Logic of Decision and Action*, University of Pittsburgh Press, Pittsburgh, Pa., p. 81-95, 1967.

Donnellan K., " Reference and Definite Descriptions", *Philosophical Review*, vol. 75, p. 281-304, 1966.

Fellbaum C. (ed.), *WordNet: An Electronic Database*, MIT Press, Cambridge, Massachusetts, 1998.

Hobbs J. R., " Ontological Promiscuity", *Proc. of the 23rd ACL*, Chicago, IL, p. 61-69, 1985.

Kamp H., " A Theory of Truth and Semantic Representation", *in* G. et al. (ed.), *Formal Methods In the Study of Language*, Mathematical Centre Tracts 135, Amsterdam, p. 277-322, 1981.

McCord M. C., " Slot Grammars", *Computational Linguistics*, vol. 6, p. 31-43, 1980.

McCord M. C., " Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars", *in* R. Studer (ed.), *Natural Language and Logic: International Scientific Symposium*, *Lecture Notes in Computer Science*, Springer Verlag, Berlin, p. 118-145, 1990.

McCord M. C., " Heuristics for Broad-Coverage Natural Language Parsing", *Proceedings of the ARPA Human Language Technology Workshop*, Morgan-Kaufmann, Princeton, New Jersey, p. 127-132, 1993.

McCord M. C., A Formal System for Slot Grammar, Technical report, IBM T. J. Watson Research Center, 2006a. RC 23976.

McCord M. C., The Slot Grammar Lexical Formalism, Technical report, IBM T. J. Watson Research Center, 2006b. RC 23977.

McCord M. C., Using Slot Grammar, Technical report, IBM T. J. Watson Research Center, 2006c. RC 23978.

McCord M. C., Bernth A., " A Metalogical Theory of Natural Language Semantics", *Linguistics and Philosophy*, vol. 28, p. 73-116, 2005.

Menzel C., A Complete, Type-free "Second-order" Logic and Its Philosophical Foundations, Technical Report n˚ CSLI Report 86-40, Stanford University, 1986.

Quirk R., Greenbaum S., Leech G., Svartvik J., *A Grammar of Contemporary English*, Longman, London, 1972.

A. Explanation of ESG parse trees

This appendix gives a brief introduction to how to read ESG parse trees.[11] As an example, below is the parse tree for *John sees Mary swim.*

```
.- subj(n)    John1(1)       noun propn sg h m gname (John#1)
o- top        see2(2,1,3,4)  verb vfin vpres sg vsubj sta (see#3V)
‘- obj(n)     Mary1(3)       noun propn sg h f gname (Mary#2)
‘- comp(binf) swim1(4,3,u)   verb vinf intrans (swim#4V)
```

The lines of the parse display are in 1-1 correspondence with the (sub-)phrases, or nodes, of the parse tree. And generally each line (or tree node) corresponds to a word of the sentence.[12] (There are exceptions to this when multiword analyses are used, and when punctuation symbols serve as conjunctions.) Slot Grammar is *dependency-oriented*, in that each node (phrase) of the parse tree has a head word, and the daughters of each node are viewed as modifiers of the head word of the node.

On each line of the parse display, you see a head word sense in the middle section, along with its logical arguments. To the left of the word sense predication, you see the slot that the head word (or node) fills in its mother node, and then you can follow the tree line to the mother node. To the right, you see the features of the head word (and of the phrase which it heads). The first feature is always the part of speech. Further features can be morphological, syntactic, or semantic. The semantic features are more open-ended, and depend on the ontology and what is coded in the lexicon. In this paper we sometimes omit less important features in order to make the parse trees fit better on the page. The last feature is the refID provided by Euphoria.

Arguments given to word sense predicates in the parse display are as follows. The first argument is just the node index, which is normally the word number of the word in the sentence. The remaining arguments correspond to the complement slots of the word sense–or rather to the fillers of those slots. They always come in the same order as the slots in the lexical slot frame for the word sense. For a verb, the first of these complement arguments (the verb sense’s second argument) is always the *logical* subject of the verb. Generally, all the arguments are *logical* arguments. Passivized expressions are “unwound” in this logical representation.

Given this, we can now explain the above parse tree (partially) as follows.

11. The description in this section is based largely on McCord (2006c) and is quoted with permission of the author.
12. Occasionally we split the line in order to accommodate the page width.

The top node has three daughters, filling the slots `subj(n)`, `obj`, and `comp(binf)`. The verb *see* has word number 2, and its subject has word number 1 (*John*). The object has word number 3 (*Mary*), and the `comp` slot is filled by word number 4, *swim*.

We note that ESG supplies the implicit subject of *swim* as word number 3, which is *Mary*.