RC24098 (W0611-025) November 7, 2006
Computer Science

IBM Research Report

Deep Analysis of Biomedical Abstracts

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

Deep analysis of biomedical abstracts

Arendse Bernth
IBM T.J. Watson Research Center
19 Skyline Dr.
Hawthorne, NY 10532
USA
arendse@us.ibm.com

Abstract

Deep analysis helps discover information that would otherwise go undetected. We report on the application of semantic typing, deep parsing, discourse understanding, and resolution of hidden arguments for information discovery in the medical domain. The focus is on relation extraction, which provides the groundwork for discovering truly new relations that are based on information in the documents, but with no explicit connection. Our concern is with relations that require deep linguistic and semantic processing. Preliminary small-scale evaluation shows promising results, with an improvement in precision from 0.4000 to 0.9155, and recall from 0.1412 to 0.7647, compared with a shallow-method baseline.

1 Introduction

This paper reports on a new project in the domain of information discovery from medical abstracts. In order to discover relevant information, extraction of relations such as those between genes and the effects of drugs is crucial. Some relations can be extracted by shallow means; others require deep linguistic and semantic processing, and still others require domain knowledge and inference. Our focus in this paper is the extraction of relations that require deep linguistic and semantic processing.

Not only *extracting* the relations is important, but also representing them in a way suitable for inference.

In this paper we describe the extraction of relations in the medical domain using a hybrid approach consisting of named-entity recognition, deep parsing, and discourse-semantic processing, including coreference, resolution of implicit arguments, and most-plausible semantics. The result is production of a normalized semantic representation, conveniently represented in a database indexed by extended entities and to be used in inference in the future.

To illustrate the kind of relations we are aiming at handling in this paper, consider the example text and relation pairs in (1).[1] The relations are shown in the normalized semantic representation. Each relation is represented as a flat predicate-argument structure in a list form, with the relation (predicate) as the first element, followed by its arguments. Each element of the list represents a discourse entity, represented by a unique referent ID, such as `activate#7V`. The semantic representation is described in detail in [Bernth, 2004] and [Bernth, 2006]. For ease of reading, we provide a gloss of the relations in English.

[1] The example is from [Yakushiji *et al.*, 2001], and the relations extracted the ones identified in that paper, even though it is not clear whether the system they report on actually extracts these relations or not. The relations shown in this paper are actual output from our system.

```
(1) 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.

Relations extracted:
((activate#7V Phorbol_Ester#5 protein_kinase#9))
A phorbol ester activates the protein kinase.

((modify#25V Phorbol_Ester#5 l_kappa_b#21))
The phorbol ester modifies L kappa B.

((dissociate#11V Phorbol_Ester#5 complex#13))
The phorbol ester dissociates a cytoplasmic complex.

((cause#23V Phorbol_Ester#5 dissociate#11V))
The phorbol ester causes the cytoplasmic complex to dissociate.
```

The above relations are available in the sentence without *inference*, but based on deep *linguistic* analysis. This deep analysis includes resolution of implicit (and long-distance) arguments such as the "subject" (*an active phorbol ester*) and "object" (*protein kinase*) of *activation*, and the subject of *modifying* (*an active phorbol ester*), as described in [Bernth, 2006]. Without resolving the implicit arguments, information would go undetected.

To see where all this is leading, consider the two sentences from two separate documents, shown in (2) and (3), respectively, along with an interesting subset of the relations extracted.

```
(2) Importantly, bone loss was almost completely prevented by p38 MAPK inhibition.

    Relations extracted:
    inhibit#25V < ((prevent#5V inhibit#25V bone_loss#1G) (inhibit#25V u p38_MAPK#3))
    p38 MAPK inhibition prevents bone loss.

(3) Thus, our results identify DLC as a novel inhibitor of the p38 pathway and provide
    a molecular mechanism by which cAMP suppresses p38 activation and promotes apoptosis.

    Relations extracted:
    DLC#10 < ((inhibit#25V DLC#10 pathway#11))
    DLC inhibits p38 pathway.

    suppress#18V < ((suppress#18V cAMP#13 activate#14V) (activate#14V u p38#2))
    cAmp suppresses p38 activation.
```

The relations shown above in (2) and (3) are crucial for discovering new information (with suitable degree of uncertainty) such as what is listed in (4).

```
(4) (a) DLC prevents bone loss.
    (b) cAMP prevents bone loss.
```

In order to derive the information in (4a) we need to know the following:

- That DLC inhibits the p38 pathway.
- That p38 MAPK inhibition prevents bone loss.
- That *inhibition of the p38 pathway* entails *p38 MAPK inhibition*.

What we've got is the following:

- For"DLC inhibits p38 pathway":
  `DLC#10 < ((inhibit#25V DLC#10 pathway#11))`
  which is reasonable.
- For "p38 MAPK inhibition prevents bone loss":
  `inhibit#25V < ((prevent#5V inhibit#25V bone_loss#1G)`
  `(inhibit#25V u p38_MAPK#3))`
  which is reasonable.
- For "*inhibition of the p38 pathway* entails *p38 MAPK inhibition*". If we had
  `DLC#10 < ((inhibit#25V DLC#10 p38_MAPK#3)` instead of
  `DLC#10 < ((inhibit#25V DLC#10 pathway#11)`
  we would have the answer. In other words, if we had the real-world knowledge that inhibiting the pathway for X inhibits X, we would have it.

In order to derive the information in (4b) we need to know the following:

- That cAMP suppresses p38 activation.
- That p38 MAPK inhibition prevents bone loss.
- That suppressing activation entails inhibition.

What we have is:

- For "cAMP suppresses p38 activation":
  `activate#14V < ((suppress#18V cAMP#13 activate#14V)`
  `(activate#14V u p38#2))`
- For "p38 MAPK inhibition prevents bone loss":
  `inhibit#25V < ((prevent#5V inhibit#25V bone_loss#1G)`
  `(inhibit#25V u p38_MAPK#3))`

Thus, the relations extracted provide the groundwork for discovering truly new relations that are based on information in the documents, but with no explicit connection. Other obvious applications would be question-answering or extraction of sentences where the relations are found.

The steps involved in getting from the input document to the normalized semantic representation of the relations are as follows:

- Named-entity recognition
- Deep parsing
- Semantic analysis:
  - Coreference resolution
  - Filling in implicit arguments
  - Producing entity-oriented logical forms (EOLFs)
- Identification of "interesting" relations.

In the following, we shall look at each of these steps in turn, using the example in (1) to illustrate each step. Section 2 describes the role of the named-entity recognition, section 3 describes the deep parsing, and section 4 the semantic analysis. Section 5 addresses the issue of extracting the interesting relations.

In sections 6 and 7 we describe two applications of the relation extraction. Section 6 describes annotating relations in a UIMA environment, and section 7 describes a prototype question-answering system, created mainly as a development tool. The main purpose is to conveniently illustrate that interesting information has been discovered, distilled and made available in a usefully normalized form. Question-answering is an easy-to-understand way of illustrating the value of coreference, normalizing linguistic variation, and resolution of implicit arguments for discovering the wealth of implicit information for our sample sentence.

Section 8 reports on a preliminary evaluation and section 9 gives our conclusion.

2 Named-entity recognition

As the first step, the document is annotated with medical terminology. At this point we are interested in drugs and genes, as well as the subtree of the MeSH ontology that deals with neoplasms: C04. Currently we are using James Cooper's annotator that annotates a wide selection of drugs and genes, and all MeSH terms. Fig. 2 shows an example of the output of this annotator.[2]

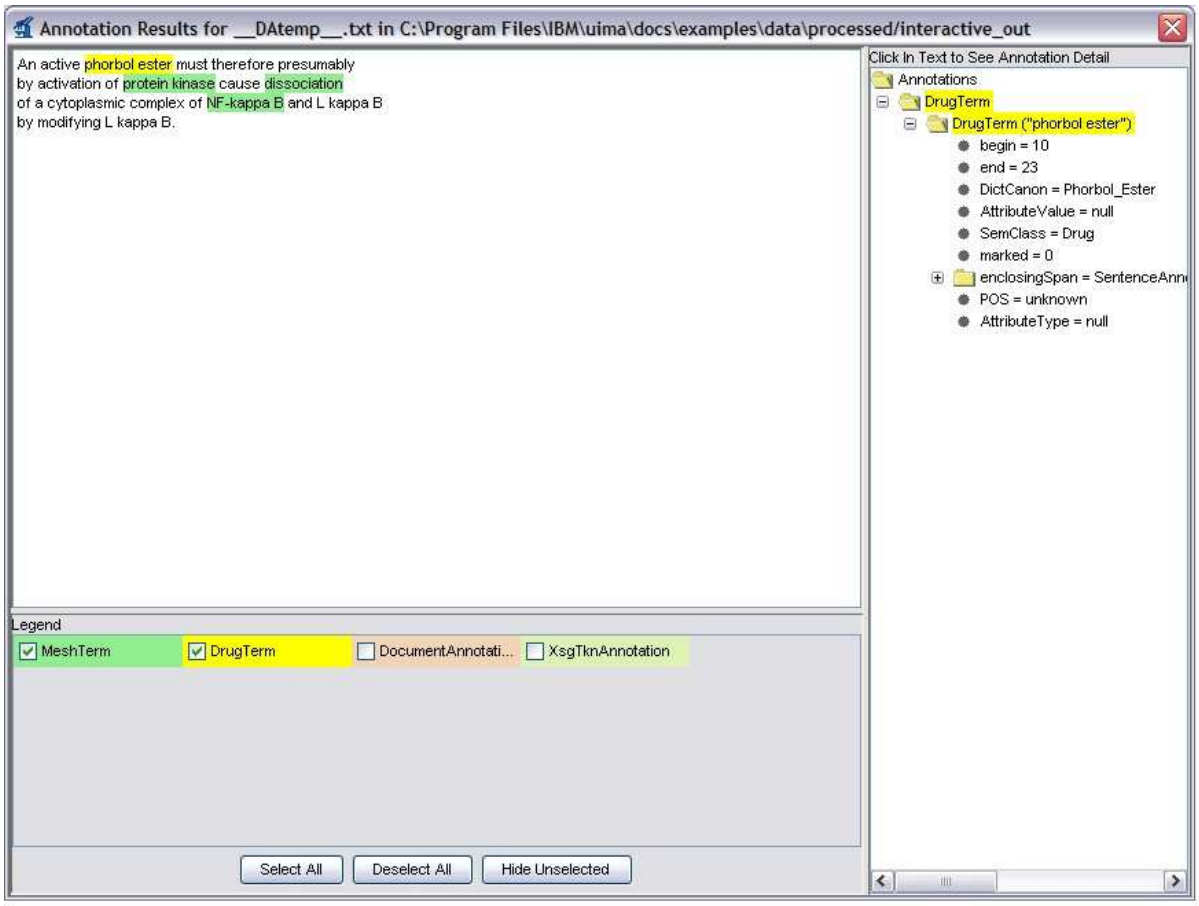

Figure 1: Named-entity recognition

The advantage of using named-entity recognition is two-fold. Most important are the semantic types provided. We know what is a drug, gene etc. This is particularly important since what drives the extraction of *interesting* relations is that at least one of the arguments is a drug, gene, or relevant MeSH term. Furthermore, since the annotator recognizes a number of synonyms and provides a canonical form of genes and drugs, some cases of noun-noun coreference are taken care of.

Secondly, chunking noun phrases helps the parser. The point is made in [Goertzel *et al.*, 2006] that many parsers have problems with the medical terms that often involve unusual combinations of special characters and numbers. Whereas the parser in question *is* able to handle such tokens, properly recognized terminology provides a flatter structure, which is easier for the semantic analysis component to interpret. Thus, we get the best of both worlds: A deep parse that is not unnecessarily cluttered by individually tokenized term parts.

[2] *phorbol ester* is actually not a drug, but this doesn't detract from the overall value of named-entity recognition and the overall scheme presented in this paper.

3 Deep parsing

The second step is deep parsing.

As pointed out by [Yakushiji *et al.*, 2001], broad-coverage, general-purpose parsing is needed for information extraction due to the great variety of ways relations can be expressed in natural language. In addition, *deep* parsing gives us access to relations that would otherwise go undetected.

For the deep parsing we use the English Slot Grammar (ESG) [McCord, 1980, McCord, 1990, McCord, 1993, McCord, 2006a, McCord, 2006b, McCord, 2006c].

It has been claimed [McDonald *et al.*, 2004] that parsers generally have problems with poor coverage as well as overgeneration. And in the case of shallow parsers there tend to be problems with long-distance relations.

However, ESG is a broad-coverage, rule-based parser. It has been measured automatically against the Penn Treebank (PTB), and achieved a score similar to the best statistical parsers, in spite of the facts that ESG was not specially trained on the PTB, and there are apples-and-oranges problems in matching ESG parses to the style of the PTB. In addition, ESG delivers deeper analyses than typical statistical parsers trained on the PTB.

For our purposes, it is particularly important that ESG gives us information about both near and long distance relations, and in some cases provides us with implicit arguments. In addition, a number of the concerns raised in [Yakushiji *et al.*, 2001], such as embedding, are addressed by the use of a good broad-coverage parser. ESG handles the potential efficiency issues mentioned by [Yakushiji *et al.*, 2001] by pruning of intermediate parse results during parsing, but is nevertheless quite accurate as mentioned above.

In (5) we show the parse for our example sentence. In order to accommodate the page width, most lines have been split into two or more lines; each node is indicated by its word number on the left.

We notice that *phorbol ester* has its semantic type (drug) and canonical form (Phorbol_Ester) from the named-entity recognizer marked (word 3); likewise *L kappa B* (word 20) has its semantic type (gene) marked. Furthermore, ESG supplies the implicit subject of the nonfinite verb *modifying* (word 22), identified as node 3, corresponding to *phorbol ester*.

```
(5) 1 'An'  ndet(dt)    parseFrameSpan: [0, 2] a ss: a(1) wordSpan: [0, 2] det sg indef
    2 'active'  nadj(a)     parseFrameSpan: [3, 9] active#1A active1
       ss: active1(2, aobj: u) wordSpan: [3, 9] adj  Euph: active#1A
  3 'phorbol ester'  subj(n)     parseFrameSpan: [0, 23] Phorbol_Ester#5 phorbol ester
     ss: phorbol ester(3) wordSpan: [10, 23] noun propn sg  type=Drug
     dictCanon=Phorbol_EsterdictCanonEnd  Euph: Phorbol_Ester#5
4 'must'  top(nop)   parseFrameSpan: [0, 179] must1 ss: must1(4, subj: 3, auxcomp: 11)
   wordSpan: [24, 28] verb vfin vpres sg vsubj
  5 'therefore'  vadv(av)    parseFrameSpan: [29, 38] therefore#27Adv therefore1
     ss: therefore1(5) wordSpan: [29, 38] adv  Euph: therefore#27Adv
  6 'presumably'  vadv(av)    parseFrameSpan: [39, 49] presumably#29Adv presumably1
     ss: presumably1(6) wordSpan: [39, 49] adv  Euph: presumably#29Adv
  7 'by'  vprep(p)     parseFrameSpan: [51, 82] by1 ss: by1(7, objprep: 8)
     wordSpan: [51, 53] prep pprefv
    8 'activation'  objprep(n)     parseFrameSpan: [54, 82] activate#7V activation1
       ss: activation1(8, nobj: 9) wordSpan: [54, 64] noun cn sg  Euph: activate#7V
      9 'of'  nobj(n)     parseFrameSpan: [65, 82] of1 ss: of1(9, objprep: 10)
         wordSpan: [65, 67] prep pprefn nonlocp
        10 'protein kinase'  objprep(n)     parseFrameSpan: [68, 82] protein_kinase#9
           protein kinase ss: protein kinase(10) wordSpan: [68, 82] noun propn sg
           Euph: protein_kinase#9
  11 'cause'  auxcomp(binf)  parseFrameSpan: [83, 179] cause#23V cause1
     ss: cause1(11, subj: 3, obj: 12, iobj: u) wordSpan: [83, 88] verb vinf
     Euph: cause#23V
    12 'dissociation'  obj(n)     parseFrameSpan: [89, 155] dissociate#11V
       dissociation1 ss: dissociation1(12, nobj: 13) wordSpan: [89, 101] noun cn sg
       Euph: dissociate#11V
      13 'of'  nobj(n)     parseFrameSpan: [103, 155] of1 ss: of1(13, objprep: 16)
          wordSpan: [103, 105] prep pprefn nonlocp
          14 'a'  ndet(dt)    parseFrameSpan: [106, 107] a ss: a(14)
              wordSpan: [106, 107] det sg indef
          15 'cytoplasmic'  nadj(a)     parseFrameSpan: [108, 119] cytoplasmic#3A
              cytoplasmic1 ss: cytoplasmic1(15) wordSpan: [108, 119] adj
              Euph: cytoplasmic#3A
        16 'complex'  objprep(n)     parseFrameSpan: [106, 155] complex#13 complex2
           ss: complex2(16, nobj: 17, nobj: u) wordSpan: [120, 127] noun cn sg
           Euph: complex#13
          17 'of'  nobj(n)     parseFrameSpan: [128, 155] of1 ss: of1(17, objprep: 19)
              wordSpan: [128, 130] prep pprefn nonlocp
              18 'NF-kappa B'  lconj(n)     parseFrameSpan: [131, 141] NF-kappa B
                 ss: NF-kappa B(18) wordSpan: [131, 141] noun propn sg
            19 'and'  objprep(n)     parseFrameSpan: [131, 155] and#17 and1 ss: and1(19)
                wordSpan: [142, 145] noun propn pl cord type=Gene  Euph: and#17
              20 'kappa'  rconj(n)     parseFrameSpan: [146, 155] l_kappa_b#19 l kappa b1
                 ss: (l1 kappa b1)(20) wordSpan: [148, 153] noun propn sg type=Gene
                 Euph: l_kappa_b#19
    21 'by'  vprep(p)     parseFrameSpan: [157, 179] by1 ss: by1(21, objprep: 22)
        wordSpan: [157, 159] prep pprefv
      22 'modifying'  objprep(ing)   parseFrameSpan: [160, 179] modify#25V modify1
          ss: modify1(22,  subj: 3, obj: 23, comp: u) wordSpan: [160, 169] verb ving
          Euph: modify#25V
        23 'kappa'  obj(n)     parseFrameSpan: [170, 179] l_kappa_b#21 l kappa b1
           ss: (l1 kappa b1)(23) wordSpan: [172, 177] noun propn sg type=Gene
           Euph: l_kappa_b#21
```

4 Semantic analysis

After parsing, the deep analysis is taken a step further by the discourse understanding system Euphoria [Bernth, 2002, Bernth, 2004, Bernth, 2006], which is built on top of ESG.

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.

The semantic analysis takes the form of *Entity-Oriented Logical Forms (EOLFs)*, as described in [Bernth, 2004] and [Bernth, 2006]. These EOLFs provide a normalized representation suitable for inference. A simple example of the normalization that takes place is the handling of passive constructions, which are "unwrapped" and result in the same EOLFs as the corresponding active constructions. A more complex example is the treatment of implicit arguments of deverbal nouns, described in [Bernth, 2006]. Here the nouns are normalized to verbs giving the relation implicit in the noun phrase, and having the associated logical subjects and objects, thus enabling inference to handle relations expressed by nouns and verbs similarly.

Two instances of this is found in our example; in (6), we give the relevant parts of the EOLFs for a reduced version of the sentence.

```
(6) An ester must by activation of kinase cause dissociation of a cytoplasmic complex.

activate#7V        < ((instr activate#7V cause#23V)
                      (activate#7V Phorbol_Ester#5 protein_kinase#9))
dissociate#11V     < ((cause#23V Phorbol_Ester#5 dissociate#11V)
                      (dissociate#11V Phorbol_Ester#5 complex#13))
```

The two deverbal nouns *activation* and *dissociation* have been normalized to verbs, and their arguments identified and filled in.

The full set of EOLFs for our example is given in (7).

```
(7) 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.

Phorbol_Ester#5  < ((active#1A Phorbol_Ester#5)
                    (activate#7V Phorbol_Ester#5 protein_kinase#9)
                    (card Phorbol_Ester#5 sing)
                    (cause#23V Phorbol_Ester#5 dissociate#11V)
                    (dissociate#11V Phorbol_Ester#5 complex#13)
                    (modify#25V Phorbol_Ester#5 l_kappa_b#21))
activate#7V       < ((instr activate#7V cause#23V)
                    (activate#7V Phorbol_Ester#5 protein_kinase#9))
active#1A         < ((active#1A Phorbol_Ester#5))
and#17            < ((and#17 NF-kappa_B#15 l_kappa_b#19))
cause#23V         < ((instr activate#7V cause#23V)
                    (cause#23V Phorbol_Ester#5 dissociate#11V)
                    (instr modify#25V cause#23V))
complex#13        < ((and complex#13) (dissociate#11V Phorbol_Ester#5 complex#13)
                    (card complex#13 sing) (cytoplasmic#3A complex#13))
cytoplasmic#3A    < ((cytoplasmic#3A complex#13))
dissociate#11V    < ((cause#23V Phorbol_Ester#5 dissociate#11V)
                    (dissociate#11V Phorbol_Ester#5 complex#13))
l_kappa_b#19      < ((card l_kappa_b#19 sing) (and#17 NF-kappa_B#15 l_kappa_b#19))
l_kappa_b#21      < ((modify#25V Phorbol_Ester#5 l_kappa_b#21)
                    (card l_kappa_b#21 sing))
modify#25V        < ((instr modify#25V cause#23V)
                     (modify#25V Phorbol_Ester#5 l_kappa_b#21))
presumably#29Adv  < ((presumably#29Adv u))
protein_kinase#9  < ((activate#7V Phorbol_Ester#5 protein_kinase#9)
                     (card protein_kinase#9 sing))
therefore#27Advc  < ((therefore#27Adv u))
```

5 Extracting interesting relations

After named-entity recognition and deep analysis, we are finally ready to extract the interesting relations. As opposed to semantic, but template-based systems such as those reported on in [McDonald *et al.*, 2004], we are not limited to a predefined set of relations. In some sense *all* of the Euphoria output is "relations"; the challenge is then to select the *interesting* parts. We follow e.g. [Liang *et al.*, 2006] and [Goertzel *et al.*, 2006] in letting the extraction be driven by the *arguments* of the relations, rather than by the relations. Thus, we extract all relations that have at least one argument in the domain of interest, *viz.* drugs, genes, and diseases.

What gets extracted as "interesting" for our example is shown in (8).

```
(8) Phorbol_Ester#5 < ((active#1A Phorbol_Ester#5)
                       (activate#7V Phorbol_Ester#5 protein kinase#9)
                       (card Phorbol_Ester#5 sing) (cause#23V Phorbol_Ester#5 dissociate#11V)
                       (dissociate#11V Phorbol_Ester#5 complex#13)
                       (modify#25V Phorbol_Ester#5 l kappa b#21))
    activate#7V     < ((instr activate#7V cause#23V)
                       (activate#7V Phorbol_Ester#5 protein kinase#9))
    l kappa b#21    < ((card l kappa b#21 sing)
                       (modify#25V Phorbol_Ester#5 l kappa b#21))
```

6 Relation extraction in a UIMA environment

One application of the relation extraction that we have just begun is extracting the relations in a UIMA environment. The relations represented as EOLFs are completely "document agnostic". *I.e.*, they are completely normalized and have little trace of the input document. For this reason, it was necessary to expand the referent IDs to encode such things as character offsets.

This means, that instead of having a referent ID like `activate#7V`, it is necessary to have an ID like `activate#7V[8,activation,54,64]`, indicating the referent ID proper (`activate#7V`); the specific mention ID, represented as an integer (`8`); the specific mention in the document `activation`; and the character offsets for delimiting the word. However, it is still important to keep the database indexed by the shorter referent ID, in order to facilitate lookup.

The resulting database hence has entries like this:

```
(9) Phorbol_Ester#5 < ((active#1A[2,active,3,9] Phorbol_Ester#5[6,phorbol_ester,10,23])
                       (activate#7V[8,activation,54,64]
                            Phorbol_Ester#5[6,phorbol_ester,10,23]
                            protein_kinase#9[10,protein_kinase,68,82])
                       (card Phorbol_Ester#5[6,phorbol_ester,10,23] sing)
                       (cause#25V[26,cause,83,88]
                            Phorbol_Ester#5[6,phorbol_ester,10,23]
                            dissociate#11V[12,dissociation,89,101])
                       (dissociate#11V[12,dissociation,89,101]
                            Phorbol_Ester#5[6,phorbol_ester,10,23]
                            complex#13[14,complex,120,127])
                       (modify#27V[28,modifying,160,169]
                            Phorbol_Ester#5[6,phorbol_ester,10,23]
                            l_kappa_b#21[22,L_kappa_b,170,179]))
activate#7V         < ((activate#7V[8,activation,54,64]
                            Phorbol_Ester#5[6,phorbol_ester,10,23]
                            protein_kinase#9[10,protein_kinase,68,82]))
```

Obviously, it was also necessary to write code to manage all this, both on the Euphoria and the UIMA ends.

Some screen shots of the relation extraction are shown below. Fig. 2 shows the relation `(modify#25V Phorbol_Ester#5 l_kappa_b#21)`, and fig. 3 the relation `(activate#7V Phorbol_Ester#5 protein_kinase#9)`. Exploration of the database will provide more relations.

7 Question-Answering

Another obvious application is question-answering, and a prototype Q/A system has been created, mainly as a development tool to facilitate exploration of the EOLFs. The main purpose is to conveniently illustrate that interesting information has been discovered, distilled and made available in a usefully normalized form.

In order to answer questions, we go through the following steps:

- Process input question using the same steps as listed above in sections 2, 3, and 4, *viz* named-entity recognition, parsing, and semantic analysis including coreference.
- Find answer in discourse EOLFs
- Generate English output from answer EOLF(s)

In section 7.1 we describe the approach to finding the answer, and in section 7.2 the component responsible for generating English output from the answer EOLFs is described.

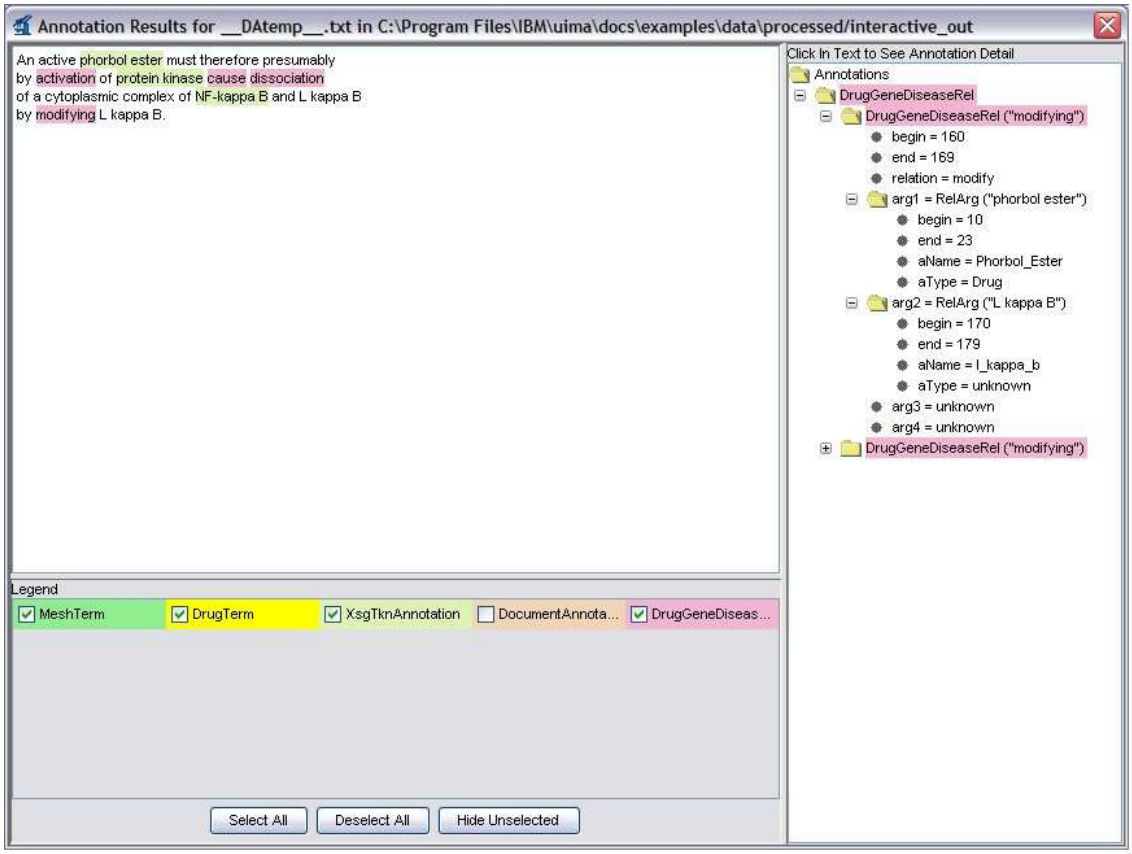

Figure 2: Relation *Phorbol ester modifies L kappa B*

7.1 Finding the answer

The EOLFs provide the basis for answering the question; currently, no domain knowledge is used, only information found directly in the EOLFs. Similarly no inference is used. The EOLF representing the input question is matched against the EOLFs and unification of arguments applied.

7.2 Generating English output

Generation is a complete research topic in itself and is not really the focus of this report, but a rudimentary generation component has been created to make it easier for the reader to understand the examples.

The generation component utilizes ESG lexicon slot-filling information to guide the generation of verb complements, which, depending on part of speech and argument structure, could be realized as either a noun or a verb; furthermore if it is realized as a verb we need to know if it should be a bare infinitive, a *to*-infinitive or a *that*-clause etc.

As a simple approach to determining the tense of verbs, we let the generated answer inherit the verb tense from the question.

Once the appropriate features of a word are determined, the LMT English generation morphology is applied to inflect the word properly, and a simple subject-verb-object sentence constructed.

There are obviously many more things that could be done to generate good English from the EOLFs. An obvious and easy thing would be to improve the focus of the answer sentence to reflect that of the question sentence. For example, in (12) the answer seems unnatural because the focus provided by the active voice clashes with the passive voice of the question.

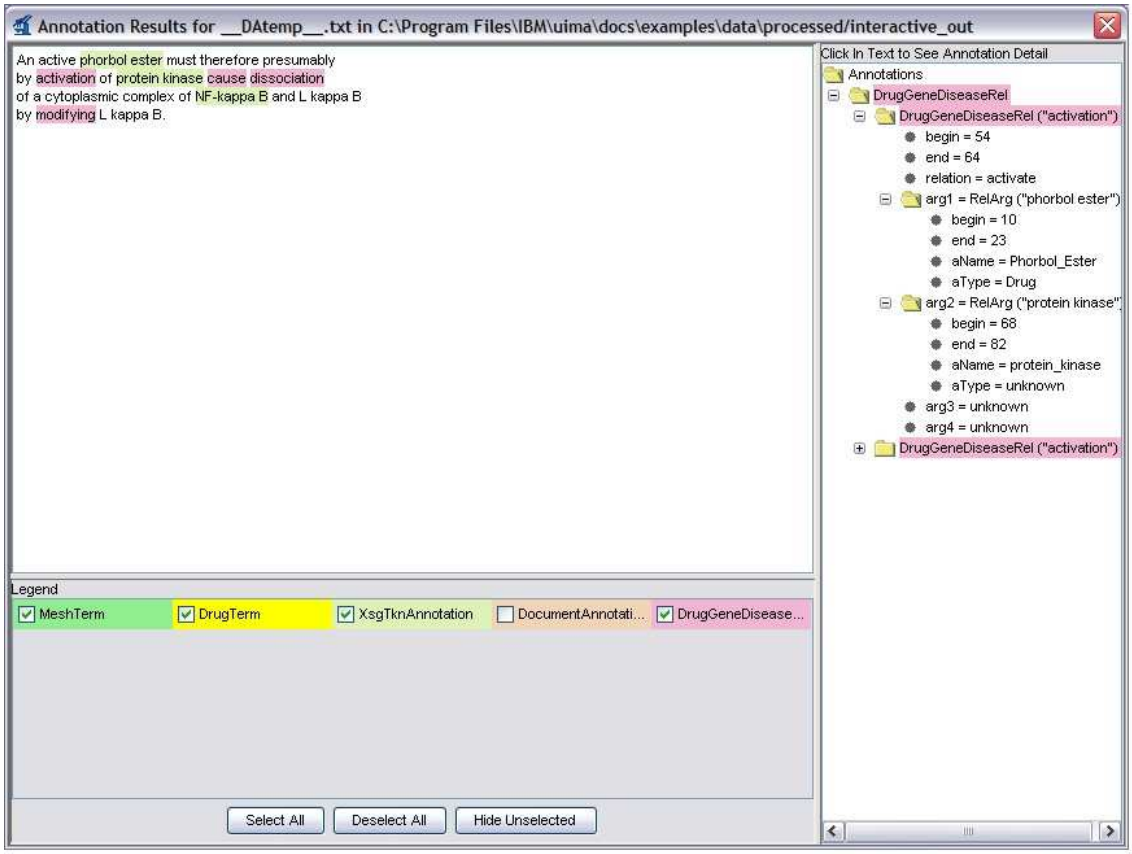

Figure 3: Relation *Phorbol ester activates protein kinase*

Another more serious and difficult issue is how much information to include in the generated text. The answer EOLF only gives the top relation, including arguments, that actually provides the answer. But more information might be in order. For example, an argument might be realized by a noun that taken by itself in English is ambiguous, but could be clarified by including some of its modifiers, such as adjectives and relative clauses. Exactly how much to include is a tricky issue.

7.3 Answering some questions

The previous sections explained the general approach. In this section we give some examples of question-answering in order to illustrate the capabilities to discover implicit information. In section 7.3.1 we illustrate the usefulness of coreference, linguistic variation, and resolution of implicit arguments for discovering the wealth of implicit information for our sample sentence, and in section 7.3.2 we ask questions of whole documents.

7.3.1 Coreference, linguistic variation, and implicit arguments

The example in (10) is repeated for convenience in comparing with examples (11) and (12). Example (11) shows the value of coreference resolution for the question; in this particular context *kinase* in example (11) is resolved to the same entity as *protein kinase* in example (10).

```
(10) 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.
```

```
(11) Question:
What activates kinase?
LF: (activate#18V what#19 protein_kinase#9)
Answer(s):
LF: ((activate#7V Phorbol_Ester#5 protein_kinase#9))
Answer in English: the active Phorbol Ester activates the protein kinase.
```

Example (12) shows the value of a normalized form; regardless of whether the question (or information in the abstract) is expressed in the active or passive voice, the same EOLF is generated, and we are able to answer the question.

```
(12) Question:
What is activated by the ester?
LF: (activate#21V ester#20 what#22)
Answer(s):
LF: ((activate#7V Phorbol_Ester#5 protein_kinase#9))
Answer in English: the active Phorbol Ester activates the protein kinase.
```

Examples (13) and (14) illustrate the value of resolving implicit arguments. In example (13) resolving the implicit argument of *modifying* allows us to discover the connection between *an active phorbol ester* and *L kappa b*, namely *modification*.

```
(13) Question:
What modifies L kappa B?
LF: (modify#23V what#24 l_kappa_b#21)
Answer(s):
LF: ((modify#25V Phorbol_Ester#5 l_kappa_b#21))
Answer in English: the active Phorbol Ester modifies an l kappa b.
```

In example (14) resolving the implicit arguments of *disassociation* allows us to discover the connection between *an active phorbol ester* and the *cytoplasmic complex*, namely *disassociation*.

```
(14) Question:
What is dissociated?
LF: (dissociate#25V u what#26)
Answer(s):
LF: ((dissociate#11V Phorbol_Ester#5 complex#13))
Answer in English: the active Phorbol Ester dissociates a cytoplasmic complex.
```

Example (15) illustrates how resolving the long-distance implicit subject of *cause* and combining it with the resolution of the arguments of *disassociation* illustrated in (14) allows us to discover the causality relation between between *an active phorbol ester* and the *cytoplastic complex*.

```
(15) Question:
What causes dissociation?
LF: (cause#27V what#28 dissociate#11V)
Answer(s):
LF: ((cause#23V Phorbol_Ester#5 dissociate#11V))
Answer in English: the active Phorbol Ester causes the cytoplasmic complex to dissociate.
```

7.3.2 PubMed example

In this section we illustrate question-answering on a whole document, a PubMed abstract. The sentences contributing to the answers are highlighted.

(16) *Aspirin-like drugs (ALD) induce calcium mobilization*, an essential component of T cell activation, but *do not induce the biosynthesis* of IL-2. To understand the extent to which ALD may mimic mitogenic stimulation, we studied cytoplasmic and nuclear signaling steps in ALD-treated T cells. We found that *ALD induce a transient activation of protein kinase (PKC)* but have no effect (in comparison to anti-CD3 antibodies) on protein tyrosine phosphorylation nor on PCL gamma 1 tyrosine phosphorylation. *ALD-induced calcium mobilization and PKC activation are independent of tyrosine protein kinase activity as shown by the lack of effect of herbimycin*, a tyrosine-protein kinase-specific inhibitor. Although we detected no IL-2 mRNA in ALD-treated cells, the nuclei of these cells contain proteins capable of binding to three regulatory sequences in the IL-2 promoter region: NFAT, NF kappa B, and AP-1. These binding activities are expressed only in activated T cells. The expression of AP-1 depended on calcium mobilization and PKC activation.

(17) Question:
What does ALD induce?
Answer(s):
aspirin+like drugs induce essential calcium mobilization.
aspirin+like drugs do not induce a biosynthesis.
aspirin+like drugs induce activation of protein.

Question:
What shows that calcium mobilization is independent [of kinase activity]?
Answer(s):
a lack of effect of herbimycin shows the calcium mobilization and protein to be independent.

8 Evaluation

We have done a preliminary evaluation of how well Euphoria extracts relations in the biomedical domain.

What we are evaluating is extraction of two relations, `inhibit` and `induce`, each assumed to have two arguments, for cases where at least one of the arguments is "interesting", as defined above in section 2. In addition, we have added the MeSH subtree for genes (G14.330) as "interesting" in order to get more results. For evaluation purposes we make the assumption that the named-entity recognition is perfect, both with respect to precision and recall, as our concern is not really to evaluate the quality of the named-entity recognition but rather the impact of deep processing over shallow methods. In particular, we are interested in getting the argument structure of the relation correct. Furthermore, at this point, the interface between Euphoria and UIMA for displaying the extracted relations is not being evaluated, as work has just only begun on exploring the Euphoria database in the UIMA environment.

As mentioned in [Ahlers *et al.*, 2007], it is common to use co-occurrence of words for extracting relations between words. However, as we are interested in not only the fact that these words are somehow related, but also that the argument structure of the relation is correct, we need to be a little more sophisticated for our baseline. Hence our baseline is as follows. A part-of-speech tagger with stemming is assumed, as well as the named-entity recognizer. The baseline looks for simple occurrences of the pattern `Subject-NP` $\cdots$ `Verb` $\cdots$ `Object-NP`, where the `Verb` indicates the relation, and the arguments are found on each side of the verb with the (deep) subject assumed to be to the left of the verb and the (deep) object assumed to be to the right of the verb. Since the relation extraction is driven by the occurrence of an "interesting" argument, the interesting argument nearest to the verb is assumed to be the subject or object depending on which side of the verb it occurs on, and the remaining argument is the nearest NP on the other side of the verb.

We hand-evaluated the extraction of the two relations `inhibit` and `induce` without the use of synonyms on a corpus of 10 medical abstracts (with a total of 29 relations) with the results shown in table 1.

Each relation consists of three parts: the relation and its two arguments. Each part counts equally in our evaluation. Hence the total number of items to be extracted consists of the number of relations multiplied

by three, and a correctly extracted relation where both arguments are correct counts as three items correctly extracted. If only one of the arguments is correct, we count two correct results out of the possible three. If both arguments are incorrect, we count the total result for that relation as incorrect, since it was a requirement that at least one of the arguments should be "interesting".

	Precision	**Recall**
Baseline	0.4000	0.1412
Euphoria	0.9155	0.7647

Table 1: Evaluation results.

As can be seen, both precision and recall improved significantly.

9 Conclusion

We have reported on progress in finding relations between drugs, genes and diseases using named-entity recognition, deep parsing and deep semantic analysis. The relation extraction is driven by the named entities rather than by a restricted list of relations, and we have illustrated the need for deep analysis to extract and normalize relations with examples that show the variety of surface forms that give rise to relations. Preliminary evalaution shows promising results.

References

[Ahlers *et al.*, 2007] Caroline B. Ahlers, Marcelo Fiszman, Dina Demner-Fushman, François Michel Lang, and Thomas C. Rindflesch. Extracting semantic predications from Medline citations for pharmocogenomics. In *Pacific Symposium on Biocomputing 12*, pages 209–220, Hawaii, 2007.

[Bernth, 2002] Arendse Bernth. Euphoria – A reference resolution system for machine translation. Technical Report RC22627, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, November 2002.

[Bernth, 2004] Arendse Bernth. Euphoria semantic analysis. Technical Report RC23396, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, November 2004.

[Bernth, 2006] Arendse Bernth. Implicit predicate arguments and discourse. Technical Report RC 24046, IBM T.J. Watson Research Center, 19 Skyline Dr., Hawthorne, NY 10532,USA, September 2006. Submitted to the special issue on Computational Approaches to Discourse and Document Processing in the journal Traitement Automatique des Langues.

[Goertzel *et al.*, 2006] Ben Goertzel, Hugo Pinto, Ari Heljakka, Izabela Freire Goertzel, Mike Ross, and Cassio Pennachin. Using dependency parsing and probabilistic inference to extract relationships between genes, proteins and malignancies implicit among multiple biomedical research abstracts. In *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL06*, pages 104–111, New York City, 2006. Association for Computational Linguistics.

[Liang *et al.*, 2006] Jisheng Liang, Thien Nguyen, Krzysztof Koperski, and Giovanni Marchisio. Ontology-based natural language query processing for the biological domain. In *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL06*, pages xxx–xxx, New York City, 2006. Association for Computational Linguistics.

[McCord, 1980] Michael C. McCord. Slot Grammars. *Computational Linguistics*, 6:31–43, 1980.

[McCord, 1990] Michael C. McCord. Slot Grammar: A system for simpler construction of practical natural language grammars. In R. Studer, editor, *Natural Language and Logic: International Scientific Symposium, Lecture Notes in Computer Science*, pages 118–145. Springer Verlag, Berlin, 1990.

[McCord, 1993] Michael C. McCord. Heuristics for broad-coverage natural language parsing. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 127–132, Princeton, New Jersey, 1993. Morgan-Kaufmann.

[McCord, 2006a] Michael C. McCord. A formal system for Slot Grammar. Technical report, IBM T. J. Watson Research Center, 2006. RC 23976.

[McCord, 2006b] Michael C. McCord. The Slot Grammar lexical formalism. Technical report, IBM T. J. Watson Research Center, 2006. RC 23977.

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

[McDonald *et al.*, 2004] Daniel M. McDonald, Hsinchun Chen, Hua Su, and Byron B. Marshall. Extracting gene pathway relations using a hybrid grammar: the Arizona Relation Parser. *Bioinformatics*, 20:18:3370–3378, 2004.

[Yakushiji *et al.*, 2001] Akane Yakushiji, Yuka Tateisi, Yusuke Miyao, and Jun-Ichi Tsujii. Event extraction from biomedical papers using a full parser. In *Pacific Symposium on Biocomputing 6*, pages 408–419, Hawaii, 2001.