Natural Language Engineering **1** (1): 1–38. Printed in the United Kingdom

A Comparison of Parsing Technologies for the Biomedical Domain

Claire Grover, Mirella Lapata, and Alex Lascarides
Division of Informatics
The University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW, UK
{C.Grover,M.Lapata,A.Lascarides@ed.ac.uk}

(*Received 2002*)

Abstract

This paper reports on a number of experiments which are designed to investigate the extent to which current NLP resources are able to syntactically and semantically analyse biomedical text. We address two tasks: parsing a real corpus with a hand-built wide-coverage grammar, producing both syntactic analyses and logical forms; and automatically computing the interpretation of compound nouns where the head is a nominalisation (e.g., *hospital arrival* means an arrival at hospital, while *patient arrival* means an arrival of a patient). For the former task we demonstrate that flexible and yet constrained 'pre-processing' techniques are crucial to success: these enable us to use part-of-speech tags to overcome inadequate lexical coverage, and to 'package up' complex technical expressions prior to parsing so that they are blocked from creating misleading amounts of syntactic complexity. We argue that the XML-processing paradigm is ideally suited for automatically preparing the corpus for parsing. For the latter task, we compute interpretations of the compounds by exploiting surface cues and meaning paraphrases, which in turn are extracted from the parsed corpus. This provides an empirical setting in which we can compare the utility of a comparatively deep parser vs. a shallow one, exploring the trade-off between resolving attachment ambiguities on the one hand and generating errors in the parses on the other. We demonstrate that a model of the meaning of compound nominalisations is achievable with the aid of current broad-coverage parsers.

1 Introduction

A growing body of research focuses on the processing, mining and extraction of biomedical knowledge from digital repositories of scientific literature such as MedLine (Hersh *et al.*, 1994). MedLine is a collection of biomedical abstracts maintained and supported by the U.S. National Library of Medicine[1] which contains approximately 10 million abstracts, and approximately 40,000 abstracts are added each month. Although MedLine is a valuable resource that allows scientists to access and

[1] http://www.nlm.nih.gov/

retrieve articles of interest, most of the information it contains is not represented in a structured format (e.g., in the form of a database) but instead in the form of natural language text. The rapid increase of novel information being added to MedLine means that hand-constructed databases and ontologies, despite their usefulness, cannot be considered exhaustive or complete. And the information available in texts like MedLine must be retrieved using automatic methods that not only access and process biomedical text efficiently but also are able to discover novel facts about medical data.

The use of computational linguistic techniques for automatically extracting information from biomedical texts (in particular from MedLine) has received increasing attention lately (Andrade & Valencia, 1998; Blaschke *et al.*, 1999; Pustejovsky *et al.*, 2001). Much of the reported work focuses either on information retrieval (Proux *et al.*, 2000; Fukuda *et al.*, 1998; Iliopoulos *et al.*, 2001) or on the detection and extraction of relations, for example between proteins and cell-types or between proteins and associated diseases (Rindfleisch *et al.*, 2000; Blaschke *et al.*, 1999; Sekimizu *et al.*, 1998; Cracen & Kumlien, 1999; Pustejovsky *et al.*, 2002; Humphreys *et al.*, 2000; Rosario & Hearst, 2001; Yakushiji et al; 2001)

Processing medical abstracts is challenging from the perspective of both syntax and semantics. Ambiguities at all levels of linguistic processing are in abundance and are hard to resolve; for example, coordination, ellipsis and complex nominals (which are used as meaning compression devices) are all common place and typically problematic for state-of-the art parsers. Furthermore, over 46% of sentences feature one or more of the following: complex equations (e.g., *chi2 = 13.1, p less than 0.001*), units of measurement (e.g., *10 mIU/ml*), numbers (e.g., *9.3 +/− 0.7*) and drug/chemical/substance names or formulae (e.g., *alpha,beta-methylene ATP*). If left unchecked, parsing such expressions engenders unnecessary syntactic complexity, and this militates against automatic knowledge discovery. Robust semantic interpretation of medical text poses additional challenges. While domain-specific knowledge bases like the UMLS metathesaurus (Humphreys *et al.*, 1998) are useful for certain interpretation tasks, such resources are not exhaustive. Some form of learning is therefore necessary, in order to complement the gaps and idiosyncrasies in the available resources.

In this paper, we report a number of experiments which are designed to investigate whether it is possible to use current state of the art NLP resources to syntactically and semantically analyse MedLine abstracts. Because of the syntactic and semantic complexity of medical text, many current information extraction systems employ tools (e.g. parsers, named-entity recognisers) or ontologies that have been specifically developed for the biomedical domain (Andrade & Valencia, 1998; Pustejovsky *et al.*, 2002; Pustejovsky *et al.*, 2001). The systems described in (Yakushiji et al; 2001) and (Proux *et al.*, 2000) contain domain specific components for named entities such as genes and proteins but they reuse general purpose parsers which are not specifically tuned to the domain. Our first suite of experiments follows this methodology in investigating whether it is possible to use state of the art NLP resources which aren't specifically developed for the biomedical domain to syntactically and semantically analyse MedLine abstracts. We first describe work on parsing these abstracts using a hand-crafted grammar which provides both syntactic and semantic analyses. Such grammars typically have insufficient coverage over real data to be of use in practical applications; this is largely due to inadequate lexical coverage. We address this problem via a suite of NLP tools which pre-process the data prior to parsing, and in particular we exploit part-of-speech (POS) tag information (rather than domain-specific lexicons) to overcome the lack of lexical coverage within the grammar.

Our second suite of experiments investigates the extent to which state-of-the-art NLP technology can be used to perform the task of interpreting compound nouns where the head noun is a deverbal head. This involves computing the semantic relation between the modifier and the head noun; for example, we aim to predict that in *patient arrival* the patient is the *agent* of the arriving event, whereas in *hospital arrival* the hospital is the *destination* of the arrival event.

We chose this task for several reasons. First, compound nominalisations are highly productive in the medical text genre and as they are frequently used as devices for 'compressing meaning' (Marsh, 1984). In a random sample of 50 sentences, 72% contained noun compounds, yielding an average of 1.4 compounds per sentence; 35% of these were compound nominalisations, i.e., on average one compound nominalisation for every two sentences. The high degree of productivity of compound nominalisations means that one cannot assume that a given compound is to be found in existing on-line domain knowledge sources, such as the UMLS lexicon or metathesaurus. Even if a given lexicon were to include linguistic (both syntactic and semantic) information about various *classes* of productive compounds (e.g., that a noun denoting a human can be combined with a nominalisation of a verb that takes animate subjects to form a compound where the semantic relation between the modifier and the head is that the modifier is the *agent* of the event associated with the head), one would still need to predict which *token* compounds that are present in the corpus but absent from the lexicon belong to which class. Either way, there is a need to tackle the task of compound noun interpretation using corpus-based methods.

The second reason for choosing the task of interpreting compound nominalisations is that it involves acquiring semantic information that is *linguistically implicit* (cf. the semantic relations mentioned above, which are required for specifying the meanings of *hospital arrival* and *patient arrival*). Indeed, interpreting compound nouns is often analysed in the linguistics literature in terms of (impractical) general purpose reasoning with pragmatic information such as real world knowledge (e.g., (Hobbs *et al.*, 1993)). Even if a hand-crafted grammar achieved perfect precision and coverage it wouldn't provide a complete description of the content that is conveyed in the corpus. Performing the task of compound noun interpretation can therefore be viewed as a complementary task to our first experiments above, of providing semantic analyses of sentences in the corpus from a hand-crafted grammar. And we aim to show that in spite of the relative difficulty of the task of compound noun interpretation—as evidenced by the traditional role of pragmatics—existing NLP technologies are sufficient for performing the interpretation task automatically. We utilise linguistically-principled assumptions in obtaining features to be used in a machine learning paradigm. We exploit meaning paraphrases and surface syntactic cues in the corpus to estimate the relation of a compound head and its modifier when the former is a nominalisation; and we aim to show that the syntactic information required can be obtained from the medical abstracts 'automatically' through parsing.

More specifically, our model of compound noun interpretation involves estimating the most likely grammatical relation between a noun and verb. For example, on the basis of information gathered from the corpus (via parsing) we can infer that *patient* is more likely to be the subject of the verb *arrive* than part of its *at*-PP complement, the difference being the likelihood of seeing *the patient arrived* vs. *arrive at the patient* in the corpus. We needed to utilise a parser so as to parse the corpus and acquire these grammatical relations automatically. This interpretation task therefore gave us the opportunity to compare two parsers: Abney's Cass partial parser (Abney, 1996) and Carroll & Briscoe's Tag Sequence Grammar (TSG) parser (Carroll & Briscoe, 2002).[2] There is a clear trade-off during parsing between on the one hand resolving syntactic ambiguities so as to obtain more complete syntactic information but potentially generating errors in the process, and on the other hand leaving such ambiguities unresolved so that the parse is more partial, but less error prone. Abney's Cass parser on average makes fewer decisions among syntactic attachment ambiguities than the TSG parser, and our goal was to investigate the effects of these different trade-offs when performing interpretation tasks on the corpus. We were particularly interested in how these trade-offs were affected when other linguistic resources were used during training, such as taxonomic information. We not only compare the effect of syntactic information gathered from different types of parsers but we also experiment with two different types of taxonomies: the UMLS meta-thesaurus (Humphreys *et al.*, 1998), which is a specialised knowledge base for the medical domain and WordNet (Miller *et al.*, 1990), a general-purpose lexical taxonomy. We aim to demonstrate that in spite of the challenges in processing biomedical text, one can acquire automatically models of a relatively difficult interpretation task by exploiting current NLP technology.

The remainder of this paper is organised as follows: Section 2 gives a brief overview of MedLine abstracts and discusses some of the challenges that biomedical texts pose for NLP technology; Section 3 describes how our corpus of medical abstracts was pre-processed using several XML-based techniques and Section 3.3 assesses their impact on parsing. Section 4 reports our experiments on the interpretation of nominalisations while comparing the effect of different parsers and taxonomies on the task.

[2] One reason why we did not use the hand-crafted grammar from the first experiment—namely the Alvey Natural Language Tools (ANLT) grammar—is that it lacks the robustness required for this task (see Section 3.3). The TSG is a relatively shallow grammar compared to the ANLT grammar, but deeper than Abney's Cass parser.

.I 309357
.U
91188323
.S
Spine 9107; 16(2):185-9
.M
Adult; Bone Screws; Case Report; Female; Fracture Fixation, Internal; Fractures, Stress/*CO/SU; Human; Lumbar Vertebrae/*IN; Male; Middle Age; Nomenclature; Spondylolysis/*CO; Support, Non-U.S. Gov't.
.T
Stress fracture of the lumbar pedicle. Case reports of "pediculolysis" and review of the literature.
.P
JOURNAL ARTICLE; REVIEW; REVIEW OF REPORTED CASES.
.W
Cases of lumbar pedicle stress fractures are described and the term "pediculolysis" introduced. The condition may be bilateral or, more commonly, may occur in association with contralateral spondylolysis. A method of direct repair with pedicle screw fixation is described.
.A
Gunzburg R; Fraser RD.

Fig. 1. *An Example from the OHSUMED Corpus*

2 The MedLine Abstracts

The work reported here uses the OHSUMED corpus of MedLine abstracts (Hersh *et al.*, 1994) which contains 348,566 references from 270 journals taken from the years 1987–1991. Each reference in the corpus has a number of attributes including the abstract itself and they are coded up in the way illustrated in Figure 1.

The abstracts are contained in the `.W` field though many references do not contain an abstract. The total number of abstracts in the corpus is 233,443 while the total number of words in the abstracts is 38,708,745—thus the average length of an abstract is 166 words. The abstracts contain approximately 1,691,383 sentences with an average length of 22.9 words. (Our notion of 'word' here is defined by the output of the tokeniser described in Section 3.)

The language in the corpus is often highly technical and contains a high number of instances of complex equations, numerical expressions, chemical formulae, etc., as illustrated in the following examples.

(1) Saved blood had a higher haemoglobin concentration (17.3 v. 13.1 g dl-1; P less than 0.001), a higher 2,3-diphosphoglycerate concentration (5.3 v. 1.1 mmol litre-1; P less than 0.00001), higher white cell count (17.1 X 10(9) litre-1 v. 4.1; P less than 0.00001), higher pH (7.5 v. 6.6; P less than 0.00001) and a more physiological potassium concentration (5.4 v. 8.8 mmol litre-1; P less than 0.00001) than donor blood.

(2) The uptake of 3.44% ID in rat heart at 1 min postinjection for [99mTc]CDO-MeB versus 3.03% for 201TI indicates high extraction of [99mTc]CDO-MeB by the myocardium.

(3) During the first 24 hours after addition of rTNF, there was a decrease in intracellular ATP content in the CEM/V line but not in the CEM line.

(4) TSH and ATP were weaker agonists compared to CC, since maximal doses of TSH (100-500 mU/ml) and ATP (100-500 microM) increased [Ca2+]i by 40-70% over basal levels, compared to a 2- to 4-fold increase in [Ca2+] induced by maximal doses of CC (10-50 microM). The TSH-induced increase in [Ca2+]i was transient, returning to basal levels within 1-2 min after application of the agonist.

(5) When N-formyl-L-methionyl-L-leucyl-L-phenylalanine (fMLP) was injected intravenously at 10 micrograms.kg-1, lung RBC content dropped by 14.7 +/- 1.8% (SE; n = 10), indicating a reduced lung blood volume, ALBev rose to $15.0 \pm 3.2\%$ of the initial albumin vascular content, and the circulating PMN were sequestered by $9.2 \pm 1.7\%$.

(6) Technetium-99m-CDO-MeB [Bis[1,2-cyclohexanedione-dioximato(1-)-O]-[1,2-cyclohexanedione dioximato(2-)-O]methyl-borato (2-)-N,N’,N”,N”’,N””,N””’)-chlorotechnetium) belongs to a family of compounds generally known as boronic acid adducts of technetium dioxime complexes (BATOs).

The sentences in (1)–(6) are fairly extreme examples of technical language and the sentences in the abstract in Figure 1 are perhaps more typical. However, the language does contain significantly high proportions of technical expressions as well as high numbers of compounds. To gain a feel for the extent of the problem, we chose 50 sentences at random from the middle part of the corpus and hand-classified substrings within them.

The notion of ‘technical expression’ is not easily definable and thus it is hard to quantify the frequency of such strings. However, the sample of 50 sentences does give some sort of indication. It contained: 23 upper-case alphabetic abbreviation-like ‘words’ (*FSH*, *MCGN*); 7 mixed case ones (*mRNA*); 16 multi-word expressions containing abbreviations (*IFN gamma*); 12 formulaic expressions containing numeric characters and units of measurement (*less than 10 mIU/ml*); and 20 other numerical expressions including dates, durations and percentages.

The notion of ‘compound’ has a more clear-cut definition. As explained in Section 1, compounding as a linguistic compression strategy is pervasive and when we counted the numbers of noun compounds in the sample of 50 sentences, we found 70 examples (average of 1.4 per sentence). This count is just for true noun compounds but there are other kinds of compound-like complex nominals which occur just as frequently. For example, there are many examples of sequences composed of adjectives and nouns which cannot easily be categorised as straightforward adjectival modification of a nominal head: (7) shows some true noun compounds from our sample of 50 sentences, while (8) shows some compound-like sequences from the same sample, which contain adjectives as well as nouns (adjectives italicised). The bracketings indicate the structure of the complex nominal and demonstrate that the adjective forms part of the compound and is not simply an adjunct of the entire nominal.

(7)
a. amino acid sequence
b. muscle fibers
c. blood pressure
d. thyrotropin level

(8)
a. ((*paediatric* oncology) clinic)
b. (((human HL-60 *myeloid* leukemia) cell) line)
c. (lung (*inflammatory* response))
d. (((*fresh* gas) flow) rates)

In Section 6 we describe our method of interpreting noun-noun compounds where the head noun is a deverbal nominalisation. This kind of compound occurs very frequently and in the sample of 50 sentences, 25 of the compounds (approx. 35%, i.e. on average one every two sentences) were headed by a deverbal nominalisation. Some of these are shown in (9):

(9)
a. drug abusers
b. sodium excretion
c. lipopolysaccharide synthesis
d. amino acid deprivation
e. ciprofloxacin resistance
f. digoxin withdrawal

3 Pre-processing the MedLine Abstracts

In this section we describe the various stages of pre-processing that we have performed in preparation for parsing. By pre-processing we mean identification of word tokens and sentence boundaries and other lower-level processing tasks such as part-of-speech (POS) tagging and lemmatisation. These initial stages of processing form the foundation of our work with MedLine abstracts and we build on them for a variety of higher level tasks. In Section 3.3 we describe a processing pipeline where initial tokenisation is succeeded by a level of recognition of technical entities such as drug names, formulae, etc. in preparation for full deep parsing, while in Section 4 we describe tokenisations to identify compounds and to mark up verbal stems on deverbal nominalisations.

Our processing paradigm is XML-based. As a mark-up language for NLP tasks, XML is expressive and flexible yet constrainable. Furthermore, XML-based tools for NLP applications lend themselves to a modular, pipelined approach to processing whereby linguistic knowledge is computed and added as XML annotations in an incremental fashion. In our work with the OHSUMED corpus, the key components of our pipelines are the programs distributed with the LT TTT and LT XML toolsets (Grover et al., 2000; Thompson et al., 1997). We have also successfully integrated non-XML public-domain tools into our pipelines and incorporated their output into the XML mark-up using the LT XML program *xmlperl* (McKelvie, 2000). Note that our pipeline architecture is similar to the architecture embodied in the GATE system (Cunningham *et al.*, 2002) in that disparate NLP processes can be plugged together to perform successive layers of processing. The major difference is that XML is fundamental to our approach whereas GATE is build around the TIPSTER architecture.

The core program in our pipelines is the LT TTT program *fsgmatch*, a general purpose transducer which processes an XML input stream and rewrites it using rules provided in a hand-written grammar file, where the rewrite usually takes the form of the addition of XML mark-up. Typically, *fsgmatch* rules specify patterns over sequences of XML elements and use a regular expression language to identify patterns inside the character strings (PCDATA) which are the content of elements. For example, the following rule for decimals such as ".25" builds a word (a W element) out of two character sequences (S elements). The rule is searching, first for an S element which contains the string "." as its content, and second for an S element which has been identified as a cardinal (C='CD', e.g. any sequence of digits). When these two character sequences are found, they are wrapped in a W element with the attribute C='CD' (targ_sg).

(10) <RULE name="decimal" targ_sg="W [C='CD'] ">
<REL match="S/# ~ ^[\ .]$"></REL >
<REL match="S [C='CD'] "></REL>
</RULE>

3.1 Word Level Processing

The first step in processing OHSUMED is a conversion from its original format to an appropriate XML format. We do this by using a Perl program to wrap an initial TEXT element around the input and then applying an fsgmatch grammar which converts the original delimiters into XML elements. The resulting structure of a RECORD element can be seen in Figure 2.

Each step in a pipeline can be thought of as a distinct module so that pipelines can be configured to different tasks. An early task is the identification of word tokens within abstracts, which we accomplish using a two-stage process. First, sequences of characters are bundled into S (sequence) elements using *fsgmatch*. For each class of character a sequence of one or more instances is identified and the type is recorded as the value of the attribute C (UCA=upper case alphabetic, LCA=lower case alphabetic, WS=white space etc.):

(11) *Arterial PaO2 as measured by*

<S C='UCA'>A</S><S C='LCA'>rterial</S><S C='WS'> </S>
<S C='UCA'>P</S><S C='LCA'>a</S><S C='UCA'>O</S>
<S C='CD'>2</S><S C='WS'> </S><S C='LCA'>as</S>
<S C='WS'> </S><S C='LCA'>measured</S>
<S C='WS'> </S><S C='LCA'>by</S>

```
<RECORD>
<ID>309357</ID>
<MEDLINE-ID>91188323</MEDLINE-ID>
<SOURCE>Spine 9107; 16(2):185-9</SOURCE>
<MESH>
Adult; Bone Screws; Case Report; Female; Fracture Fixation, Internal; Fractures,
Stress/*CO/SU; Human; Lumbar Vertebrae/*IN; Male; Middle Age; Nomenclature;
Spondylolysis/*CO; Support, Non-U.S. Gov't.
</MESH>
<TITLE>
Stress fracture of the lumbar pedicle. Case reports of “pediculolysis” and review of the
literature.
</TITLE>
<PTYPE>
JOURNAL ARTICLE; REVIEW; REVIEW OF REPORTED CASES.
</PTYPE>
<ABSTRACT>
<SENTENCE><W P=‘NNS’ LM=‘case’>Cases</W> <W P=‘IN’>of</W>
<W P=‘JJ’ LM=‘lumbar’>lumbar</W> <W P=‘NN’ LM=‘pedicle’>pedicle</W>
<W P=‘NN’ LM=‘stress’>stress</W> <W P=‘NNS’ LM=‘fracture’>fractures</W>
<W P=‘VBP’ LM=‘be’>are</W> <W P=‘VBN’ LM=‘describe’>described</W>
<W P=‘CC’>and</W> <W P=‘DT’>the</W> <W P=‘NN’ LM=‘term’>term</W>
<W P=‘”’>“</W><W P=‘NN’ LM=‘pediculolysis’>pediculolysis</W><W P=‘”’>”</W>
<W P=‘VBD’ LM=‘introduce’>introduced</W><W P=‘.’>.</W></SENTENCE>
<SENTENCE><W P=‘DT’>The</W><W P=‘NN’ LM=‘condition’>condition</W>
<W P=‘MD’ LM=‘may’>may</W> <W P=‘VB’ LM=‘be’>be</W>
<W P=‘JJ’>bilateral</W><W P=‘CC’>or</W><W P=‘,’>,</W>
<W P=‘RBR’>more</W> <W P=‘RB’>commonly</W><W P=‘,’>,</W>
<W P=‘MD’ LM=‘may’>may</W> <W P=‘VB’ LM=‘occur’>occur</W>
<W P=‘IN’>in</W> <W P=‘NN’ LM=‘association’>association</W>
<W P=‘IN’>with</W> <W P=‘JJ’>contralateral</W>
<W P=‘NN’ LM=‘spondylolysis’>spondylolysis</W><W P=‘.’>.</W></SENTENCE>
<SENTENCE><W P=‘DT’>A</W> <W P=‘NN’ LM=‘method’>method</W>
<W P=‘IN’>of</W> <W P=‘JJ’>direct</W>
<W P=‘NN’ LM=‘repair’>repair</W> <W P=‘IN’>with</W> <W P=‘NN’
LM=‘pedicle’>pedicle</W> <W P=‘NN’ LM=‘screw’>screw</W>
<W P=‘NN’ LM=‘fixation’>fixation</W> <W P=‘VBZ’ LM=‘be’>is</W>
<W P=‘VBN’ LM=‘describe’>described</W><W P=‘.’>.</W></SENTENCE>
</ABSTRACT>
<AUTHOR>
Gunzburg R; Fraser RD. </AUTHOR>
</RECORD>
```


Fig. 2. *A sample from the XML-marked-up OHSUMED Corpus*

Here, all characters including white space and newline are contained in S elements which become building blocks for the next call to *fsgmatch* where words are identified. An alternative approach would find words in a single step but our two-step method provides a cleaner set of word-level rules which are more easily modified and tailored to different purposes; modifiability is critical since the definition of what is a word can differ from one subsequent processing step to another. Once the word-level grammar has applied, the S mark-up can be discarded, as in Figure 2.

The LT TTT toolset includes a program called *ltpos* which is a combined sentence boundary disambiguator and part-of-speech (POS) tagger (Mikheev, 1997). The point at which we use this program depends on the particular task: for the deep parsing described in Section 3.3 it is used early on, immediately after word identification, to disambiguate sentence boundaries and then a second call is made late in the pipeline to perform POS tagging after all the higher-level tokenisation is complete. In the task described in Section 5, *ltpos* is used to provide the tags needed by the Cass chunker and is applied soon after word-level tokenisation has been completed. Note that the tagset used by *ltpos* is the Penn Treebank tagset (Marcus *et al.*, 1993).

In the processing so far, each module has used one of the LT TTT or LT XML programs which are sensitive to XML structure. There are, however, a large number of tools available from the NLP community which could profitably be used but which are not XML-aware. We have integrated some of these tools into our pipelines using the LT XML program *xmlperl*. This is a program which makes underlying use of an XML parser so that rules defined in a rule file can be directed at particular parts of the XML tree-structure. The actions in the rules are defined using the full capabilities of Perl. This gives the potential for a much wider range of transformations of the input than *fsgmatch* allows and, in particular, we use Perl's stream-handling capabilities to pass the content of XML elements out to a non-XML program, receive the result back and encode it back in the XML mark-up. One example of this method is our integration of Minnen et al.'s (2000) *morpha* lemmatiser. Here, the PCDATA content of verbal and nominal W elements is passed to the lemmatiser and the lemma that is returned is encoded as the value of the attribute LM. A sample of the output from the pipeline is shown in Figure 2.

3.2 Tokenisation above the Word Level

A major strength of our XML-based pipeline approach is the modularity it affords and the way in which layers of processing can be gradually applied in order to add or modify mark-up by increments. It may seem that this pipeline approach does not allow ambiguities to be encoded and that it therefore encourages early commitment. However, most kinds of ambiguity can easily be represented in XML through the choice of an underspecified representation (cf. much current work on underspecification in computational linguistics) and apparent cases of structural ambiguity can also often be represented in a well-designed annotation scheme. Where structural ambiguity can really not be avoided, 'standoff' mark-up (Ide *et al.*, 2000) can provide a solution.

For many NLP activities, pre-processing often seems to be limited to word-level tokenisation and tagging, but our paradigm offers the possibility of adding further layers of processing prior to the end application. In Section 3.3 we describe how extensive use of further pre-processing techniques can make a significant contribution to the use of a hand-coded syntactic and semantic grammar for deep parsing.

In section 5 we describe the use of two shallow parsing methods, and here too, it proved useful to perform some layers of tokenisation on top of the word tokenisation in order to package up a number of pervasive types of sequences such as numerical expressions, and also to deal with issues such as hyphenation and parentheses. In the remainder of this section we describe these extra layers of processing.

To deal with a wide-range of numerical expressions, we re-used a general purpose *fsgmatch* grammar which identifies and marks up multi-word numbers:

(12) <W C='CD'>Fifty-five</W>

<W C='CD'>–0.552</W>

<W C='CD'>25 million</W>

<W C='CD'>One hundred and forty six</W>

<W C='CD'>1.5 million</W>

A second above-word-level layer of tokenisation that we performed for the deep and shallow parsing tasks concerns hyphenation. In the initial tokenisation into words, hyphenated words were split into word and hyphen tokens, in order that later processes could determine which should be treated as single word tokens and which should be left split. In the *fsgmatch* grammar used here, hyphenated strings were wrapped as one word if the subwords were entirely alphabetic. In addition, hyphenated strings representing durations were treated as single words:

(13) <W C='HYW'>long-term</W>

<W C='HYW'>anti-HBc-positive</W>

<W C='HYW'>30-min</W>

<W C='HYW'>5-hr</W>

A final step that we took at this stage was to remove parenthesised material from our input to both the deep and shallow parsing tasks. We did this by marking up parentheticals using an *fsgmatch* grammar and then by removing these elements in their entirety using an *xmlperl* rule file. Thus our pipeline would remove the parenthetical material in the following examples:

(14)
a. Atrial natriuretic peptide (ANP) levels were measured .....
b. HCl was infused at a constant rate of 25 mmol/h until the bicarbonate concentration decreased less than 26 mmol/L, or until the pH decreased less than 7.35 (initial pH greater than 7.40) ....
c. PaCO2 during precordial compression was highly correlated with PetCO2 (r = .89) ....
d. Five developed pneumonitis (fatal in three);

The assumption behind the removal of parentheticals is that their contents are likely to be less central than unparenthesised material. In some cases, this assumption will turn out to be incorrect but in the context of the experiments we are

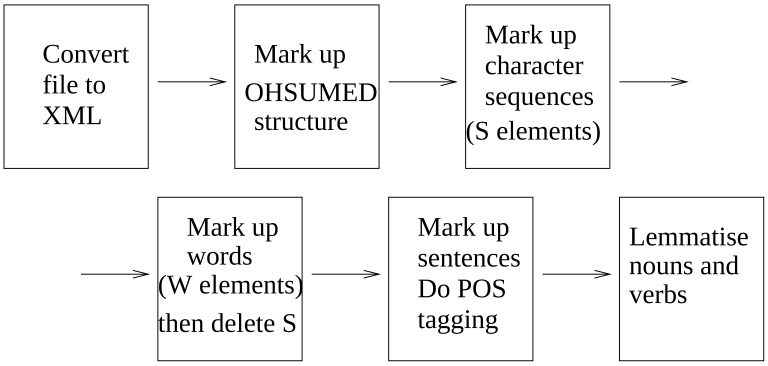

Fig. 3. Pipeline architecture

reporting here, we felt that this was a simplifying step which would not impact significantly on the task at hand.

The modular structure of the pipeline can be represented graphically as shown in Figure 3.

3.3 Robust Deep Parsing

In the first of our parsing experiments with OHSUMED, we have been attempting to improve the coverage of a hand-crafted, linguistically motivated grammar which provides full-syntactic analysis paired with logical forms. The grammar and parsing system we use is the wide-coverage grammar, morphological analyser and lexicon provided by the Alvey Natural Language Tools (ANLT) system (Carroll et al. 1991, Grover et al. 1993). Our aim was to increase coverage up to a reasonable level so that it can be of use to practical applications. The processing involved in this experiment is described in more detail in (Grover & Lascarides, 2001).

The ANLT grammar is a feature-based unification grammar based on the GPSG formalism (Gazdar *et al.*, 1985). In this framework, lexical entries carry a significant amount of information including subcategorisation information. Thus the practical parse success of the grammar is significantly dependent on the quality of the lexicon. The ANLT grammar is distributed with a large lexicon which was derived semi-automatically from a machine-readable dictionary (Carroll & Grover, 1988) and, while this provides a core of commonly-occurring lexical entries, there remains a significant problem of inadequate lexical coverage. If we try to parse OHSUMED sentences using the ANLT lexicon and no other resources, we achieve very poor results (2% coverage) because most of the medical domain words are simply not in the lexicon and there is no 'robustness' strategy built into ANLT. Rather than pursue the labour-intensive course of augmenting the lexicon with domain-specific lexical resources, we have developed a solution which does not require that new lexicons be derived for each new domain type and which has robustness built into the strategy. Furthermore, this solution does not preclude the use of specialist lexical resources such as UMLS if these can be used to achieve further improvements in performance.

Our approach relies on the sophisticated XML-based tokenisation and POS tagging described in the previous section and it builds on this by combining POS tag information with the existing ANLT lexical resources. We preserve POS tag information for content words (nouns, verbs, adjectives, adverbs) since this is usually reliable and informative and we dispose of POS tags for function words (complementizers, determiners, particles, conjunctions, auxiliaries, pronouns, etc.) since the ANLT hand-written entries for these are more reliable and are tuned to the needs of the grammar. Furthermore, unknown words are far more likely to be content words, so knowledge of the POS tag will most often be needed for content words.

Having retained content word tags, we use them during lexical look-up in one of two ways. If the word exists in the lexicon with the same basic category as the POS tag then the POS tag plays a 'disambiguating' role, filtering out entries for the word with different categories. If, on the other hand, the word is not in the lexicon or if it is not in the lexicon with the relevant category, then a basic underspecified entry for the POS tag is used as the lexical entry for the word, thereby allowing the parse to proceed. For example, if the following partially tagged sentence is input to the parser, it is successfully parsed.

(15) We studied_VBD the value_NN of transcutaneous_JJ carbon_NN dioxide_NN monitoring_NN during transport_NN

Without the tags the parse would fail since the word *transcutaneous* is not in the ANLT lexicon. Furthermore, *monitoring* is present in the lexicon but as a verb and not as a noun. For both these words, ordinary lexical look-up fails and the entries for the tags have to be used instead. Note that the case of *monitoring* would be problematic for a strategy where tagging is used only in case lexical look-up fails, since here it is incomplete rather than failed. The implementation of our word_tag pair look-up method is specific to the ANLT system and uses its morphological analysis component to treat tags as a novel kind of affix. See Grover and Lascarides (2001) for further details.

Another impediment to parse coverage is the prevalence of technical expressions and formulae in biomedical and other technical language, as discussed in the previous sections. For example, the following sentence has a straightforward overall syntactic structure but the ANLT grammar does not contain specialist rules for handling expressions such as *5.0+/-0.4 grams tension* and thus the parse would fail.

(16) Control tissues displayed a reproducible response to bethanechol stimulation at different calcium concentrations with an ED50 of 0.4 mM calcium and a peak response of 5.0+/-0.4 grams tension.

Our response to issues such as these is to place a further layer of processing in between the output of the initial tokenisation pipeline in Figure 2 and the input to the parser. Since the ANLT system is not XML-based, we already use *xmlperl* to convert sentences to the ANLT input format of one sentence per line with tags appended to words using an underscore. We can add a number of other processes at this point to implement a strategy of using *fsgmatch* grammars to package up technical expressions so as to render them innocuous to the parser. Thus all of the

Table 1. *Parse Results*

	Version 1	Version 2	Version 3
Parses	4 (2%)	32 (16%)	79 (39.5%)

following 'words' have been identified using *fsgmatch* rules and can be passed to the parser as unanalysable units. The classification of these examples as nouns reflects a hypothesis that they can slot into the correct parse as noun phrases but there is room for experimentation since the conversion to parser input format can rewrite the tag in any way.

(17)
<W P='NN'>P less than 0.001</W>
<W P='NN'>166 +/- 77 mg/dl</W>
<W P='NN'>2 to 5 cc/day</W>
<W P='NN'>9.1 v. 5.1 ml</W>
<W P='NN'>2.5 mg i.v.</W>

In addition to these kinds of examples, we also package up other less technical expressions such as common multi-word words and spelled out numbers (the expressions in the right-hand-side column are in the ANLT input format, as converted automatically from the expressions on the left):

(18)

<W P='CD'>thirty-five</W>	thirty-five_CD
<W P='CD'>Twenty one</W>	Twenty~one_CD
<W P='CD'>176</W>	176_CD
<W P='IN'>In order to</W>	In~order~to_IN
<W P='JJ'>in vitro</W>	in~vitro_JJ

In order to measure the effectiveness of our attempts to improve coverage, we conducted an experiment where we parsed 200 sentences taken at random from OHSUMED (see Grover and Lascarides (2001)). We processed the sentences in three different ways and gathered parse success rates for each of the three methods. Version 1 established a 'no-intervention' baseline by using the pipeline in Figure 3 to identify words and sentences but otherwise discarding all other mark-up. Version 2 addressed the lexical robustness issue by retaining POS tags to be used by the grammar in the way outlined above. Version 3 applied the full set of preprocessing techniques including the packaging-up of formulaic and other technical expressions. The parse results for these runs are shown in Table 1.

The baseline of 2% demonstrates that the problem is intractable without some kind of intervention. When we address the issue of lexical inadequacies by utilising POS tag information in Version 2 there is a highly significant increase in performance, although the 16% result is still sufficiently low to demonstrate that there are other factors apart from lexical gaps which impede parsing. The steps we have taken in

Version 3 to handle technical expressions lead to another very significant increase in performance. The final figure of 39.5% is still low but it demonstrates that our approach has made significant inroads into the problem.

Although we have achieved an encouraging overall improvement in performance, the total of 39.5% for Version 3 is not a precise reflection of accuracy of the parser. In order to determine accuracy, we hand-examined the parser output for the 79 sentences that were parsed and recorded whether or not the *correct* parse was among the parses found. Of these 79 sentences, 61 (77.2%) were parsed correctly while 18 (22.8%) were not, giving a total accuracy measure of 30.5% for Version 3. While this figure is rather low for a practical application, it is worth reiterating that this still means that nearly one in three sentences are not only correctly parsed but they are also assigned a logical form. We are confident that further development cycles (see below) will achieve an improvement in performance which will lead to a useful semantic analysis of a significant proportion of the corpus. Furthermore, in the case of the 18 sentences which were parsed incorrectly, it is important to note that the 'wrong' parses may sometimes be capable of yielding useful semantic information. For example, the grammar's compounding rules do not yet include the possibility of coordinations within compounds so that the NP *the MS and direct blood pressure methods* can only be wrongly parsed as a coordination of two NPs. However, the rest of the sentence in which the NP occurs is correctly parsed.

An analysis of some of the sentences which failed to parse shows that three main factors contribute to parse failure: tagging errors, tokenisation errors and lack of syntactic coverage of the grammar. All three of these problems can be addressed through further development cycles. The tokenisation of technical expressions was performed entirely by means of hand-coded rulesets and was by no means comprehensive. Future improvements to this component, perhaps incorporating machine-learning techniques to identify technical expressions, would bring us a long way towards realising our goal of performing full parsing of the majority of sentences in the corpus. It should also be noted that the results reported here were achived without altering the grammar, save the adaptation of the word grammar to handle word_tag pairs. It is to be expected that the grammar does lack coverage in certain cases and updates to it would also improve performance. Once coverage reaches an acceptable level, the next step would be to rank the parses using a parse selection method, such as Briscoe and Carroll's (1993) statistical LR-parsing method.

In the following we focus on the interpretation of the nominalisations. We first discuss the linguistic properties of nominalisations and previous work relating to the automatic interpretation of compound nouns. We then describe our experiments on the biomedical domain using a machine learning paradigm which is based on data obtained through parsing and conceptual information available in general purpose and specialised for the medical domain taxonomies.

4 Nominalisations

The automatic interpretation of compound nouns has been a long-standing unsolved problem for NLP. A considerable amount of effort has gone into specifying the set of semantic relations that hold between a compound head and its modifier (Levi, 1978; Warren, 1978; Finin, 1980; Isabelle, 1984). Levi (1978), for example, distinguishes two types of compound nouns: (a) compounds consisting of two nouns which are related by one of nine predicates (e.g., CAUSE relates *onion tears*, FOR relates *pet spray*, see the examples in (19)) and (b) nominalisations, which are compounds where the head noun is derived from a verb and its modifier is interpreted as an argument to the verb (see the examples in (20)–(21)).

(19)	a.	onion tears	CAUSE
	b.	vegetable soup	HAVE
	c.	music box	MAKE
	d.	steam iron	USE
	e.	pine tree	BE
	f.	night flight	IN
	g.	pet spray	FOR
	h.	peanut butter	FROM
	i.	abortion problem	ABOUT
(20)	a.	cell survival	SUBJ
	b.	breast conservation	OBJ
	c.	hospital readmissions	TO
	d.	dissolution treatment	WITH
	e.	hospital discharge	FROM
(21)	a.	airway opening	
	b.	policy implications	
	c.	knee alignment	

Nominalisations are abundant in the biomedical domain: 35% of the compounds found in the MedLine abstracts are nominalisations. The compound modifier can be the subject or object of the nominalised head (see examples (20a,b)) and in some cases the underlying verb subcategorises for a PP-object as shown in (20c)–(20e). The interpretation of nominalisations poses several challenges for current NLP systems. The relations between a head and its modifier are not readily available in the corpus and thus they must somehow be retrieved and approximated. The phenomenon is quite productive and therefore one cannot solely rely on resources that hand-code semantic information. Finally, nominalisations can be multiply ambiguous and most of the cases processed by a hypothetical semantic tagger would manifest some degree of ambiguity. Consider the examples in (21). Out of context, *airway opening* can mean that the airway is opening something (subject interpretation) or that something/someone is opening the airway (object interpretation). Similarly, *policy implications* can mean the policy implies something or that something is implied for the policy, and *knee alignment* can mean that the knee aligns with something or that something aligns with the knee.

Most approaches that have previously addressed the interpretation of compounds require large amounts of hand-crafted knowledge and place emphasis on recovering relations other than nominalisations. Most symbolic approaches are limited to a specific domain due to the large effort involved in hand-coding domain knowledge such as information about causes and their typical effects. Such accounts are distinguished in two main types: concept-based and rule-based. Under the concept-based approach each noun is associated with a concept and various slots. Compound interpretation reduces to slot filling, i.e., evaluating how appropriate concepts are as fillers of particular slots. A scoring system evaluates each possible interpretation and selects the highest scoring analysis (Finin, 1980; McDonald, 1982). Under the rule-based approach interpretation is performed by sequential rule application. A fixed set of rules are applied in a fixed order, and the first rule for which the conditions are met results in the most plausible interpretation (Leonard, 1984; Vanderwende, 1994). A variant of the concept-based approach uses unification to constrain the semantic relations between nouns represented as feature structures (Jones, 1995).

A statistical approach for the compound interpretation task was pioneered by Lauer (1995) who provided a probabilistic model of compound noun paraphrasing (e.g., *state laws* are "the laws of the state", *war story* is "a story about war", etc.). Lauer's model takes into account only prepositional paraphrases of compounds (e.g., *of*, *for*, *in*, *at*, etc.) and explicitly excludes nominalisations. The model assigns probabilities to different paraphrases using a corpus in conjunction with Roget's publicly available thesaurus. It combines the probability of the modifier given a certain preposition with the probability of the head given the same preposition, and assumes that these two probabilities are independent. Lauer's model achieves an accuracy of 47.0%.

Rosario & Hearst (2001) attempt a task similar to Lauer's for the biomedical domain. Rosario & Hearst develop their own inventory of 38 semantic relations (e.g., PROCEDURE characterises *tissue pathology*, INSTRUMENT characterises *biopsy needle*) and use it to annotate compounds extracted from MedLine. They recast compound noun interpretation in terms of a classification task and use neural networks in conjunction with UMLS, a lexical hierarchy designed specifically for the medical domain (see Section 6.3.1 for details), to perform the classification achieving an accuracy of 60%. Again nominalisations are not handled specifically but the relations SUBJECT and OBJECT (see examples (20a,b)) are included in the set of relations proposed by Rosario and Hearst.

The automatic interpretation of nominalisations has been previously addressed by Lapata (2002). Lapata's work focused on nominalisations naturally occurring in the British National Corpus (BNC, (Burnard, 1995)) and proposed a probabilistic model for the interpretation of compounds whose modifier is either the subject or direct object of the deverbal head, thus excluding cases with PP-arguments (see (20c)–(20e)). The argument relation between a deverbal head and its modifier was approximated (via parsing) by mapping the compound head to its underlying verb and counting the number of times the modifier was its subject or object. In the face of data sparseness the co-occurrence frequency of a verb and its argument was recreated using smoothing methods that either rely on corpus-internal distributional evidence (Dagan *et al.*, 1999) or on corpus-external semantic resources such as WordNet or Roget's thesaurus (Lauer, 1995; Resnik, 1993). The approach achieved an accuracy of approximately 80% on the binary classification task.

Nominalisations provide an interesting testbed for evaluating the portability of NLP resources to the biomedical domain. The estimation of the likelihood of an interpretation relies on the availability of parsed data, at least if one follows the approach put forward by Lapata (2002). Before parsing, the biomedical text must be part-of-speech tagged and tokenised. State-of-the-art parsers are frequently trained and tested on the Penn Treebank (Marcus *et al.*, 1993) and it is not clear whether they produce meaningful syntactic analyses for different domains and text genres. In addition to obtaining correct syntactic analyses, argument relations must be identified for interpreting nominalisations. In this paper we evaluate whether state-of-the-art NLP resources can be used to syntactically and semantically analyse MedLine abstracts. We evaluate these different resources against the nominalisation interpretation task. Previous work (e.g., Lapata 2001) has shown that the task is amenable to an empirical approach which relies on the combination of several resources that are typically available for domain-independent text.

In what follows we focus on processing the MedLine abstracts for obtaining data useful for the nominalisation task. Similarly to Rosario & Hearst (2001) we treat their interpretation as a classification task and experiment with different features using the C4.5 decision tree learner (Quinlan, 1993). Following Lauer (1995) and Lapata (2002) we adopt a paraphrasing task. The argument relation between a deverbal head and its modifier is approximated by the relation of the underlying verb and its arguments. In contrast to Lapata (2002) we take PP-objects into account (see (20c)–(20e)) and use verb-argument counts as features for the decision tree learner.

In order to obtain counts of argument relations we experiment with two parsers: Abney's (1996) partial parser Cass and the statistical parser developed by Briscoe and Carroll (2002). Both parsers extract argument relations, but differ in the amounts of linguistic knowledge they rely upon in order to produce syntactic analyses. Notably, Briscoe and Carroll's parser produces a full grammatical analysis, whereas Cass only identifies shallow linguistic patterns (i.e., chunks) without attempting to resolve attachment ambiguities. None of these parsers have been previously used to process biomedical text. In the following sections we describe how these parsers were employed to obtain counts for the nominalisation interpretation task and compare their performance. In cases where we find no evidence about the co-occurrence of a verb and its arguments, we recreate the missing counts using taxonomic information. We experiment with WordNet (Miller *et al.*, 1990), a general purpose resource and also with UMLS, a lexical database designed specifically for the biomedical domain.

In order to compare the relative utility among the various linguistic resources available, we in fact acquired several models for interpreting nominalisations and compared their performance on the interpretation task. These models varied along the following dimensions. First, they either used verb-argument counts acquired from Abney's (1996) parser Cass or from Briscoe and Carroll's (2002) tag sequence grammar (the TSG). Secondly, they either used WordNet or UMLS to smooth over sparse data on verb-argument counts. Third, the models either included as a parameter the nominalisation affix of the deverbal head (e.g., *-ation*, *-ment*) or they did not. And finally, the models either included information about the context of the nominalisation or they did not. This context was encapsulated in various ways: either as part-of-speech information or as word forms; and the context taken into account ranged from 1 word/POS tag to 5, occurring to the left of the nominalisation and/or to the right of it.

Section 5 describes how we built upon the pre-processing described in Section 3 to process the entire OHSUMED corpus, first with Cass and then with the TSG. Section 6 focuses on the interpretation task; it reports our machine learning experiments, and compares and evaluates our different features.

5 Extracting Grammatical Relations

In the following sections we briefly describe the two parsers we used to extract information about grammatical relations, and we assess their performance on the biomedical data.

5.1 Chunking with Cass

We first parsed our corpus of MedLine abstracts with Cass (Abney, 1996) a chunker whose main feature is the finite-state cascade technique. A finite-state cascade is a sequence of non-recursive levels: phrases at one level are built on phrases at the previous level without containing same level or higher-level phrases. Two levels of particular importance are *chunks* and *simplex clauses*. A chunk is the non-recursive core of intra-clausal constituents extending from the beginning of the constituent to its head, excluding post-head dependents (i.e., NP, VP, PP), whereas a simplex clause is a sequence of non-recursive clauses (Abney, 1996). Cass recognises chunks and simplex clauses using a regular expression grammar without attempting to resolve attachment ambiguities.

The parser comes with a large-scale grammar for English and a built-in tool that extracts predicate-argument tuples out of the parse trees that Cass produces. More specifically, the tool identifies subjects and objects as well as PPs without however distinguishing arguments from adjuncts. We consider passive verbs followed by the preposition *by* and a head noun as instances of verb-subject relations. Our verb-object tuples also include prepositional objects even though these are not explicitly identified by Cass. We assume that PPs adjacent to the verb and headed by the prepositions *about, against, as, at, between, by, for, from, in, into, of, on, through, to* or *with* are prepositional objects.

The input to the process is the entire OHSUMED corpus after it has been pre-processed as described in Section 3: this involves the pipeline in Figure 3 followed by the additional layers of processing described in Section 3.2. The output format of this tokenisation has to be converted to Cass's input format which is a non-XML file containing one word per line with tags separated by the tab character. We achieve this conversion using *xmlperl* with a simple rule file. The output of Cass and the grammatical relations processor is a list of each verb-argument pair in the corpus as shown in Table 2.

Table 2. *Verb-argument pairs obtained from Cass*

Verb	Argument Relation	
manage	OBJ	refibrillation
respond	SUBJ	psoriasis
access	TO	system
protect	AGAINST	osteoporosis
anaesthetize	WITH	oxide

5.2 Parsing with the Tag Sequence Grammar

Our second method of acquiring verb grammatical relations uses the statistical parser developed by Briscoe and Carroll (Briscoe & Carroll (2002), Carroll & Briscoe (2002)) which is an extension of the ANLT grammar development system. The statistical parser, known as the Tag Sequence Grammar (TSG) parser, uses a hand-crafted grammar where the lexical entries are POS tags rather than the word forms themselves. Thus it is strings of tags that are parsed rather than strings of words. The statistical part of the system is the parse ranking component where probabilities are associated with transitions in an LR parse table. The grammar does not achieve full-coverage but on the OHSUMED corpus we were able to obtain parses for 99.05% of the sentences. The number of parses found per sentence ranges from zero into the thousands but the system returns the highest ranked parse according to the statistical ranking method. We do not have an accurate measure of how many of the highest ranked parses are actually correct but even a partially incorrect parse may still yield useful grammatical relations data.

Carroll and Briscoe (2002) map TSG parse trees to representations of grammatical relations. (For details of the grammatical relation annotation scheme, see (Carroll *et al.*, 1998; Carroll *et al.*, 1999)). This format can easily be mapped to the same format as described in Section 5.1 to give counts of the number of times a particular verb occurs with a particular noun as its subject, object or prepositional object. The Carroll and Briscoe scheme identifies the surface subject of passives as underlying objects and in the mapping to our format we recover the *obj* relation.

The pre-processing for TSG input was less elaborate than that described in Section 3, with less above-word-level tokenisation owing to the use of a different tagger. As explained above, the TSG parses sequences of tags. However, it requires a different tagset from that produced by *ltpos*, namely the CLAWS2 tagset (Garside, 1987). To prepare the OHSUMED corpus for parsing with the TSG we therefore tagged it with Elworthy's (1994) tagger and since this is a non-XML tool we used *xmlperl* to invoke it and to incorporate its results back into the XML mark-up. Sentences were then converted to the TSG input format.

Figure 4a illustrates the number of tokens obtained for each argument relation from MedLine with Cass and the TSG. Figure 4b shows the ratio of the counts obtained by Cass to the counts obtained by the TSG. Note that the counts obtained from Cass are consistently higher than those obtained from the TSG parser. For

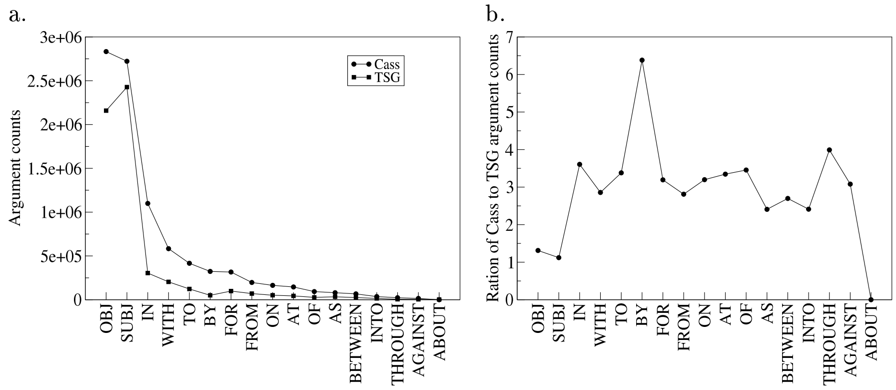

Fig. 4. Argument Relations in MedLine extracted with Cass and the TSG

prepositions this is to be expected, since the TSG parser's statistical method of ranking parses will sometimes attach PP immediately following as an adjunct where for Cass post-verbal PPs are assumed to be complements (see Section 5.1). By computing the ratio we can observe how the parsers differ for individual argument relations. The ratio for PP-arguments ranges from 2.41 to 3.99 with the exception of BY (6.38) and ABOUT (0.0003), while those for SUBJ and OBJ are 1.12 and 1.31, respectively.

5.3 Discussion

Both Cass and the TSG parser provide a shallow syntactic analysis. The former only identifies shallow linguistic patterns (i.e., chunks) without attempting to resolve attachment ambiguities, whereas the latter outputs a full syntactic analysis. There is a trade-off between choosing among various syntactic analyses and generating errors: the TSG tends on average to disambiguate and hence is more likely than Cass to generate a parse that doesn't comply with the 'right' parse; on the other hand Cass on average generates a more partial parse than TSG.

We further explored the relation between Cass and the TSG parser by comparing the obtained corpus frequencies for the different argument relations. More specifically, we explored whether there is a linear relationship between the counts obtained with Cass and the TSG parser using correlation analysis for the argument relations presented in Figure 4. The verb-argument frequencies were log-transformed. This was necessary as we carried out our analysis on log-transformed frequencies. A statistically significant correlation coefficient was obtained for the Cass and TSG counts (Pearson's $r = .766$, $N = 17$, $p < 0.05$). This indicates that despite their differences the two parsers extract related information from the biomedical domain. We further investigate their differences and similarities using a task-based evaluation paradigm, i.e., the automatic interpretation of nominalisations.

```
<W P='NN' LM='reaction' VSTEM='react'>reaction</W>
<W P='NN' LM='growth' VSTEM='grow'>growth</W>
<W P='NN' LM='control' VSTEM='control'>control</W>
<W P='NN' LM='coding' VSTEM='code'>coding</W>
```

Fig. 5. Annotated Nominalisations in OHSUMED

6 Interpreting Nominalisations

Having collected two different sets of frequency counts from the entire MedLine corpus for verbs and their arguments, we performed an experiment to discover (a) whether it is possible to reliably predict semantic relations in nominalisation-headed compounds and (b) whether the two methods of collecting frequency counts make any significant difference to the process. In the following section we describe how compound nouns in general and nominalisations in particular were annotated in our corpus. Section 6.1 describes how the data necessary for our classification task were obtained and reports an annotation study which assesses whether our class inventory can be reliably used by humans.

6.1 Data Collection

To collect data for the experiments reported below we needed to (a) mark-up deverbal nominalisations with information about their verbal stem to give nominalisation-verb equivalences and (b) to mark-up compounds in order to collect samples of two-word compounds headed by deverbal nominalisations. For the first task we combined the lemmatiser with the use of lexical resources. In a first pass we used a second call to the *morpha* lemmatiser to find the verbal stem for *-ing* nominalisations such as *screening*. Then we looked up the remaining nouns in a nominalisation lexicon which we created by combining the nominalisation list which is part of the knowledge sources provided by UMLS (2000 version) with the NOMLEX nominalisation lexicon (Macleod *et al.*, 1998). As a result of this processing, it was possible for a large proportion of the deverbal nominalisations in OHSUMED to be marked up with a VSTEM attribute whose value is the verbal stem (see Figure 5). Since this method is dependent on the completeness of the combined nominalisation lexicon, there were some cases which were missed. For example, a sample of some of the nouns which were not marked as nominalisations included *exposure* and *focus* because these were absent from the nominalisation lexicon.

To mark up compounds we developed an *fsgmatch* grammar for compounds of arbitrary length and we used this to process a subset of the first two years of the corpus. More specifically, our compound detection procedure involved not only noun sequences but also sequences composed of adjectives and nouns. Manual inspection of a sample of 1,000 candidate compound sequences showed that our *fsgmatch* grammar achieved an accuracy of 95.6%.

Table 3. *Distribution of Nominalisation Classes*

Class	Example	Frequency	
SUBJ	*age distribution*	295	(26.67%)
OBJ	*weight loss*	516	(46.65%)
WITH	*graft replacement*	27	(2.44%)
TO	*treatment response*	13	(1.17%)
ON	*knee operation*	8	(0.72%)
FOR	*nonstress test*	6	(0.54%)
IN	*vessel obstruction*	5	(0.45%)
FROM	*blood elimination*	4	(0.36%)
ABOUT	*diabetes knowledge*	3	(0.27%)
AGAINST	*seizure protection*	1	(0.09%)
BY	*aerosol administration*	1	(0.09%)
INTO	*lipid composition*	1	(0.09%)
OF	*water deprivation*	1	(0.09%)
NA	*stroke death*	177	(16.0%)
NV	*survival analysis*	48	(4.34%)

6.2 Inter-annotator Agreement

Using the LT XML program *sggrep* we extracted all sentences containing two-word compounds headed by deverbal nominalisations and from this we took a random sample of 1,000 nominalisations. These were manually disambiguated using the categories shown in Table 3. These categories denote the argument relation between the deverbal head and its modifier. We also included the categories NV (non deverbal) for nouns that are either part of a larger compound or simply adjacent without being in head modifier relationship (e.g., *survival analysis* is part of the larger term *proportional hazards survival analysis*) and NA (non-applicable) for nominalisations with relations other than the ones predicted by the verb's subcategorisation frame (e.g., in *stroke death*, *stroke* is not the argument of the verb *die*).

Before attempting to interpret nominalisations automatically we evaluated if humans can decide whether the above categories can be reliably assigned to nominalisations. Two judges were presented with 200 nominalisations (a subset of the original 1,000) and were asked to use any of the relations presented in Table 3 for the annotation task. The judges were not medical experts; they were given some simple guidelines (e.g., use the class NA when the modifier is not an argument of the compound head, use a preposition if the modifier is the object of a deverbal head which is derived from a verb subcategorising for a PP) but no prior training. The nominalisations were disambiguated in context: the judges were given the corpus sentence in which the nominalisation occurred together with the previous and following sentence. The judges were advised to consult the UMLS database (`http://umlsks4.nlm.nih.gov/`) to retrieve the meaning of unknown words.

The judges' agreement, measured using the Kappa coefficient (Cohen, 1960), was $K = .75$ ($N = 200$, $k = 2$) which translates to a percent agreement of 82.7%. This agreement was good given that the judges were not medical experts and were provided with minimal instructions. Lapata (2002) reports an agreement of .78 on the BNC for the simpler task of deciding whether a nominalisation receives a subject or an object interpretation.

6.3 Experiments

We used a machine learning approach for the disambiguation of nominalisations which explored several syntactic, semantic, and contextual features which we describe below. The different features were combined using the C4.5 decision tree learner (Quinlan, 1993). Decision trees are among the most widely used machine learning algorithms. They perform a general to specific search of a feature space, adding the most informative features to a tree structure as the search proceeds. The objective is to select a minimal set of features that efficiently partitions the feature space into classes of observations and assemble them into a tree.

In contrast to Lapata (2002) our classification is not binary but includes several argument relations (e.g., BY, FOR, TO, see Table 3). The classifier was trained and tested using 10-fold cross-validation on 1,000 manually disambiguated nominalisations. For the experiments reported in this paper we used the Weka (Witten & Frank, 2000) implementation of the C4.5 decision tree learner. Section 6.3.1 describes the inventory of features we experimented with and Section 6.3.2 reports our results.

6.3.1 Features for Interpreting Nominalisations

Frequency of Argument Relations For each candidate nominalisation our task is to predict the correct interpretation class (see Table 3). We approximate nominalisations by verb argument relations and for each candidate in the training set we record the number and types of relations extracted from the OHSUMED corpus. So, nominalisations are represented by a vector of counts of the argument relations presented in Table 3. Example 22 shows these vector representations for the compounds *graft replacement*, *cell stimulation*, and *temperature response*. The counts in 22 represent the argument relations ABOUT, AGAINST, BY, FOR, FROM, IN, INTO, OF, ON, THROUGH, TO, WITH, OBJ, and SUBJ, respectively.

(22) a. graft replacement [0, 7, 0, 0, 3, 0, 0, 0, 0, 0, 0, 24, 18, 0]
b. cell stimulation [0, 170, 2, 91, 257, 12, 0, 8, 0, 7, 7, 41, 564, 175]
c. temperature response [0, 0, 0, 0, 3, 0, 0, 0, 0, 2, 2, 0, 0, 0]

Recall that one of our goals is to directly compare Cass and the TSG parser on the nominalisation interpretation task. In order to perform this comparison, we will represent nominalisations by two types of vectors obtained from Cass and TSG counts, respectively. Note that the vectors in (22) are relatively sparse. This is partly because the OHSUMED corpus contains abstracts of medical articles; these abstracts aim to summarise and condense the information present in the main article and are abundant with compounds which are typically used as a text compression device (Marsh, 1984), i.e., to pack meaning into a minimal amount of linguistic structure.

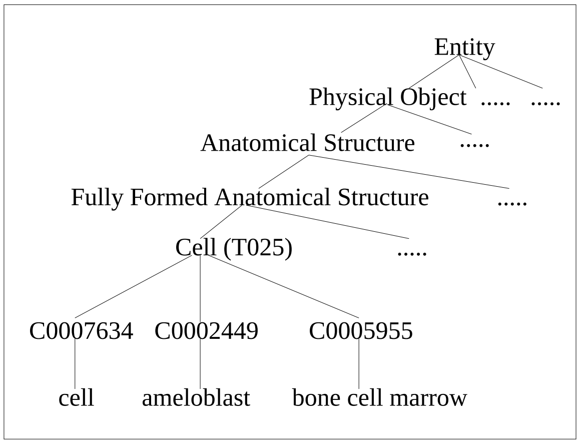

Fig. 6. A Fragment of the UMLS Metathesaurus

The occurrence of verbs is therefore sparser. This means that we can't tell in advance whether a zero count in (22) is the result of a linguistic constraint (i.e., a given verb does not take a PP object) or merely the result of insufficient evidence.

To counterbalance sparse data problems we also experimented with a simple smoothing approach. More specifically, we recreated the frequency of verb-argument relations for which we obtained a zero count using a simplified version of Resnik's (1993) measure of selectional association; this is based on relative entropy and uses a taxonomy to estimate the co-occurrence frequency of a predicate and its argument by substituting the argument with the class by which it is represented in the taxonomy.

In a nutshell, this measure replaces Resnik's information-theoretic approach with a simpler measure which makes no assumptions with respect to the contribution of a semantic class to the total quantity of information provided by a predicate about the semantic classes of its argument. It simply substitutes the argument occurring in the predicate-argument relation with the concept by which it is represented in a taxonomy and estimates predicate-argument co-occurrence frequency by counting the number of times the concept corresponding to the argument is observed to co-occur with the predicate in the corpus. Because a given word is not always represented by a single class in the taxonomy (i.e., the argument co-occurring with a predicate can generally be the realization of one of several conceptual classes), the frequency counts for a predicate-argument relation are constructed for each conceptual class by dividing the contribution from the argument by the number of classes to which it belongs.

This approach has been used by Lauer (1995) for recreating the frequencies of noun-preposition bigrams for the interpretation of compound nouns, by Lapata (2002) for the interpretation of nominalisations, and by Lapata *et al.* (2001) for modelling plausibility judgements for adjective-noun bigrams. Most previous approaches relied on taxonomic information available in WordNet (Lapata *et al.*, 2001; Lapata, 2002) or Roget's thesaurus (Lauer, 1995; Lapata, 2002) for recreating unseen predicate-argument combinations. In this paper we experiment with WordNet (Miller *et al.*, 1990) but also with the UMLS Metathesaurus (Humphreys *et al.*, 1998).

The Metathesaurus is a database of information on concepts that appear in one or more of a number of different controlled vocabularies and classifications used in the field of biomedicine. In essence, its purpose is to link alternative names and views of the same concept together and to identify useful relationships between different concepts. The UMLS Semantic Network provides a categorisation of all concepts represented in the Metathesaurus and the relationships between them. The Network contains 134 semantic types (e.g., Biologic Function Events, Organisms, Chemicals) and 54 relationships (e.g., Parent, Child, Sibling, Synonym). Words are linked to concepts and inherit high level semantic types via "isa" links. Figure 6 shows a fragment of the UMLS Metathesaurus.[3] As can be seen from Figure 6 lexical items (e.g., *cell*, *bone cell marrow*) are mapped to unique concept IDS (e.g., C0007634). The semantic type of these concepts is T025 (i.e., Cell); its hypernyms are "Fully Formed Anatomical Structure", "Anatomical Structure", "Physical Objects", and "Entity".

We recreated the frequencies of sparse verb-argument relations using both WordNet and the UMLS Metathesaurus as we wanted to see whether domain-specific resources would have an impact on the interpretation task as opposed to a general purpose taxonomy like WordNet. Example (23) shows the vector representations from (22), this time with recreated frequencies. Note that only counts for argument relations with zero frequencies are recreated. As can be seen from (23c) for some argument relations the recreated frequency is also predicted to be zero.

(23)
a. graft replacement [0.12, 7, 1.63, 0.38, 3, 0.22, 0.21, 0.41, 0.66, 0.21, 5.6, 24, 18, 6.3]
b. cell stimulation [0.16, 170, 2, 91, 257, 12, 0.33, 8, 0, 7, 7, 41, 564, 175]
c. temperature response [0, 0, 0, 1, 3, 0, 4, 1, 0, 2, 2, 4, 3.25]

To summarise, predicate argument relations were encoded in six different ways: (a) raw counts as obtained from Cass, (b) raw counts extracted from the output of the TSG, (c) recreated counts using Cass and WordNet, (d) recreated counts based on Cass and UMLS, (e) recreated TSG counts with WordNet, and (f) smoothed TSG counts with UMLS. Our experiments in Section 6.3.2 investigate the influence of these different factors.

Affixes In some cases the nominalisation affix of the compound head is indicative of whether its modifier is a subject or object. For example, head nouns with the affix *-er* typically receive an agentive interpretation (e.g., *builder* is someone who builds things) (Rappaport & Levin, 1990). Similarly, the affix *-or* or *-our* prompts a subject related interpretation (e.g., *behaviour*). We exploited the semantics of the

[3] The information displayed in Figure 6 is the result of post-processing and piecing together different sources of information that are typically kept separate in the Metathesaurus. This was necessary as we wanted to have a taxonomy comparable to WordNet.

Table 4. *Nominalisation Affixes*

Affix	Example	Frequency	
-ion	*reaction*	394	(42.09%)
-conv	*test*	282	(30.13%)
-ation	*localization*	109	(11.65%)
-ment	*impingement*	63	(6.73%)
-ing	*greeting*	25	(2.67%)
-ance	*aberrance*	19	(2.03%)
-ition	*abolition*	12	(1.28%)
-ence	*abstinence*	11	(1.17%)
-ure	*departure*	11	(1.17%)
-age	*blockage*	10	(1.06%)

nominalisation affixes by including them as features for our decision tree learner. More specifically, each candidate nominalisation head was morphologically analysed into a verb and a derivational affix. The latter was directly used as a feature for the machine learning. Morphological information about deverbal nouns was extracted from a nominalisation lexicon which was created from UMLS and NOMLEX (see Section 6.1 for details). Table 4 displays the distribution of the most frequent affixes in our data (frequency > 10). The feature "conv" represents conversions, i.e. nouns derived from verbs without the addition of an affix (Quirk *et al.*, 1985).

Context The features described above capture information about the argument relations between a deverbal head and its modifier without however taking context into account. Lapata's (2001) study showed that contextual features are important for the interpretation of nominalisation and perform as well as numeric features that are based on co-occurrence frequency. We therefore included the context surrounding the nominalisation as an additional feature for the decision tree learner. Context was encoded as lemmas or parts of speech to the left and right of the candidate nominalisations. We also varied the window size parameter between one and five words before and after the nominalisation target. Example (24) shows the contextual features for *graft replacement*. In (24b) the feature vector consists of the nominalisation and a context of five words (i.e., lemmas) to its right and five words to its left. In (24c) the lemmas are reduced to their parts of speech.

(24) a. The operative repair was accomplished by **graft replacement** of the involved segment of the aorta in all but one patient who underwent a primary repair.
b. [operative, repair, be, accomplish, by, graft, replacement, of, the, involved, segment, of]
c. [JJ, NN, VBD, VBN, IN, NN, NN, IN, DT, JJ, NN, IN]

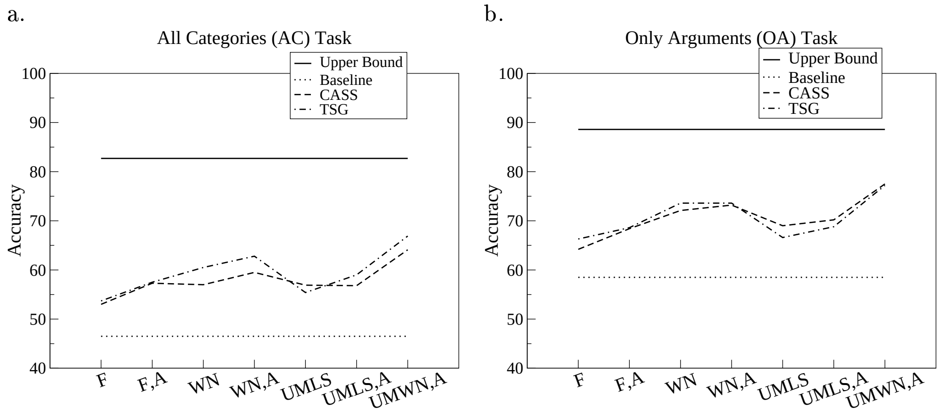

Fig. 7. Disambiguation Accuracy for Cass and TSG using raw and smoothed counts

6.3.2 Results

In this section we explore the effect of different features on the interpretation task. We report experiments on two tasks: (a) the task of predicting all classes found in the training data, including NAs and NVs (recall from Section 6.2 that these categories were reserved for compounds that were either mistakenly identified as nominalisations or did not receive an argument related interpretation) and (b) the task of predicting only nominalisations that receive argument related interpretations, i.e., excluding NAs and NVs. We called the first task AC (standing for All Categories) and the second OA (standing for Only Arguments). For both the AC and OA tasks the results are compared to the naive baseline, which (always) chooses the most frequent relation (i.e., OBJ). We also report an upper bound on disambiguation performance by measuring how well human judges agree with one another (percentage agreement) on the class assignment task. Recall from Section 6.2 that 200 instances were annotated by two judges with the categories shown in Table 3. The agreement on the AC task was 82.7%. The agreement for the OA task was 88.6%. The latter was computed after excluding from the data set instances that were classified by the judges as either NV or NA.

Figure 7a shows the performance of the decision tree learner on the AC classification task when using argument frequencies obtained with the TSG and Cass, respectively. Figure 7b reports accuracy on the OA task. For both tasks, we examine the influence of raw and smoothed frequencies on the classification task with or without affix related information. The following features were taken into account both for Cass and the TSG (see x-axis in Figures 7a,b): (a) raw frequency (F), (b) raw frequency and afix related information (F,A), (c) smoothed frequency using WordNet (WN), (d) smoothed WordNet frequency and affixes (WN,A), (e) smoothed frequency using UMLS, (f) smoothed UMLS frequency and affixes (UMLS,A), and (g) smoothed frequency using UMLS and WordNet in combination with affixes (UMWN,A).

As can be seen from Figures 7a,b the decision tree learner outperforms the naive baseline for both tasks (AC and OA), although the learner's performance is not close to the human upper bound. All features are significantly better than the baseline

Table 5. *Significance testing between* TSG *and Cass features*

	AC Task		OA Task	
Features	χ^2	p-value	χ^2	p-value
$\mathrm{F}_{\mathrm{Cass}} - \mathrm{F}_{\mathrm{TSG}}$	0.12	0.73	0.81	0.37
$\mathrm{F{,}A}_{\mathrm{Cass}} - \mathrm{F{,}A}_{\mathrm{TSG}}$	0.01	0.93	0.01	0.92
$\mathrm{WN}_{\mathrm{Cass}} - \mathrm{WN}_{\mathrm{TSG}}$	2.83	0.09	0.07	0.79
$\mathrm{WN{,}A}_{\mathrm{Cass}} - \mathrm{WN{,}A}_{\mathrm{TSG}}$	2.60	0.10	0.05	0.82
$\mathrm{UMLS}_{\mathrm{Cass}} - \mathrm{UMLS}_{\mathrm{TSG}}$	0.47	0.49	1.14	0.28
$\mathrm{UMLS{,}A}_{\mathrm{Cass}} - \mathrm{UMLS{,}A}_{\mathrm{TSG}}$	1.16	0.30	0.45	0.50
$\mathrm{UMWN{,}A}_{\mathrm{Cass}} - \mathrm{UMWN{,}A}_{\mathrm{TSG}}$	1.92	0.16	0.03	0.86

and worse than the upper bound. Significance values and pairwise comparisons (using the χ^2 statistic) for all features are given in the Appendix (see Tables 8,9).

Let us first concentrate on the AC task and the features obtained with the TSG. WN, WN,A, UMLS,A (but not UMLS) and UMWN,A significantly outperform the raw argument frequency feature F. WN is significantly better than F,A and UMLS but not UMLS,A or WN,A, whereas UMWN,A significantly outperforms all other features reaching an accuracy of 66.9%. A similar result is observed with Cass-related features. Again UMWN,A significantly outperforms all other features with an accuracy of 64.1%. F,A and WN,A are significantly better than F, but not better than WN, UMLS, or UMLS,A. In fact, there is no significant difference among the features FA, WN, WN,A, UMLS and UMLS,A. Higher accuracies are observed with the TSG rather than with Cass, however the difference is not statistically significant. Table 5 shows pairwise comparisons between Cass-related and TSG-related features using the χ^2 statistic).

We now turn to the simpler OA task. In general, higher accuracies are observed for OA than for AC (compare Figures 7a and 7b). This is not surprising given that the features we employed are particularly tailored to the nominalisation interpretation task and are not expected to be very useful for identifying noun sequences that are *not* nominalisations (i.e., NVs) or compounds that do *not* express argument relations (i.e., NAs). Similarly to the OA task, the best performing feature for both the TSG and Cass is UMWN,A achieving an accuracy of 77.2% and 77.5%, respectively. For the features estimated from the output of the TSG we observe the following (for significance values see Tables 10,11 in Appendix): all features significantly outperform the baseline, but perform significantly worse than the upper bound. The feature F performs significantly worse than any other feature. F,A performs significantly worse than WN,A and UMWN,A but it does not outperform WN, UMLS or UMLS,A. There is no significant difference between WN and UMLS or UMLS,A. WN,A significantly outperforms UMLS,A but not UMWN,A. In fact, the latter feature significantly outperforms all other features.

Similar tendencies are observed for Cass related features. All features but UMLS and UMLS,A significantly outperform the raw frequency feature F. The feature WN,A (but not WN) is significantly better than F,A and UMLS,A (but not UMLS). The

Table 6. *Argument relations (AC task)*

	AC TSG			AC Cass		
Class	Precision	Recall	F-measure	Precision	Recall	F-measure
SUBJ	0.61	0.60	0.60	0.59	0.57	0.58
OBJ	0.70	0.79	0.74	0.70	0.78	0.73
WITH	0.63	0.44	0.52	0.63	0.37	0.47
TO	0.78	0.54	0.64	0.78	0.54	0.64
ON	0.50	0.25	0.33	1	0.25	0.40
FOR	0.60	0.5	0.55	0.43	0.5	0.46
IN	0.33	0.2	0.25	0	0	0
FROM	0	0	0	0	0	0
ABOUT	1	0.67	0.80	0.5	0.33	0.40
AGAINST	0	0	0	0	0	0
BY	0	0	0	0	0	0
INTO	0	0	0	0	0	0
OF	0	0	0	0	0	0
NA	0.55	0.48	0.52	0.53	0.48	0.50
NV	0.42	0.27	0.33	0.36	0.25	0.23

feature UMLSWN,A performs significanlty better than any other feature. The two parsers do not yield significantly different results as shown in Table 5.

To summarize, we observe the following tendencies so far for both tasks. Smoothed argument counts yield better accuracies than raw frequencies when WordNet is used to recreate unseen counts, whereas this is not the case when UMLS is employed for the smoothing task. The combination of the two taxonomies outperforms all other features both for Cass and the TSG. The type of parser employed for the extraction of argument relations does not seem to have a large effect on the disambiguation task: we found no significant difference in accuracy between Cass and the TSG.

Table 6 reports precision, recall and F-measure on the AC task for individual argument relations when the learner is trained/tested with the best feature combination, i.e., UMWN,A. Table 7 shows the learner's performance on the OA task. TSG generally outperforms Cass on the AC task for all relations (including NA and NV) except for ON (see Table 6). Note that the learner has difficulty with the relations AGAINST, BY, INTO, and OF. This is not surprising given that these appear only once in the training/test corpus (see Table 3). Similar tendencies are observed for the simpler OA task (see Table 7): TSG performs better than Cass for all relations except for FOR and SUBJ. A general observation is that both Cass and the TSG are fairly good at predicting object and subject relations. Predicting PP-related arguments is relatively harder. The is due to data sparseness but also to the fact that these relations are harder to identify accurately while parsing. Recall that Cass does not distinguish between arguments and adjuncts and the TSG makes this decision stochastically.

The fact that WordNet, a general purpose taxonomy, outperforms UMLS, a biomedical thesaurus, is somewhat counterintuitive as one would expect the latter to have

Table 7. *Argument relations (OA task)*

	AC TSG			AC Cass		
Class	Precision	Recall	F-measure	Precision	Recall	F-measure
SUBJ	0.66	0.62	0.64	0.67	0.64	0.65
OBJ	0.77	0.84	0.80	0.77	0.84	0.80
WITH	0.78	0.52	0.62	0.77	0.48	0.59
TO	0.78	0.54	0.64	0.70	0.54	0.61
ON	0.60	0.38	0.46	0	0	0
FOR	0.50	0.50	0.50	0.75	0.50	0.60
IN	0.33	0.20	0.25	0.17	0.20	0.18
FROM	0	0	0	0	0	0
ABOUT	1	0.67	0.80	0.50	0.33	0.40
AGAINST	0	0	0	0	0	0
BY	0	0	0	0	0	0
INTO	0	0	0	0	0	0
OF	0	0	0	0	0	0

a larger vocabulary overlapping with the biomedical data, thus resulting in a higher performance than using WordNet. We decided to investigate the discrepancy between UMLS and WordNet further by looking at the size of the two dictionaries and their coverage on our data.

UMLS provides semantic definitions for 1,277,338 terms. Of these terms only 14% are simplex terms (i.e., one word). The majority of the terms in UMLS are either compounds (*gene product*, *nervous system*) or complex units such as *100 plus b5 8ch dt70502 cath*, *2,2-thiodisuccinic acid*, or *accident-explosion of methane*. WordNet has 94,473 entries for nouns. Of these 50.6% are simplex terms. Both Cass and TSG identify the NPs or PPs a given verb might take as arguments and from those extract only the head noun. This means that the verb-argument tuples used for recreating unseen counts do not contain any complex terms. From the simplex nouns listed in UMLS, 7.30% are attested as verbal arguments in tuples extracted by the TSG; 8.86% of the simplex UMLS nouns are found as arguments in the tuples obtained with Cass. 25.10% of the simplex WordNet nouns are arguments in TSG tuples and 25.20% are arguments in Cass tuples. These figures indicate that the argument nouns extracted either via Cass or the TSG are more likely to be found in WordNet than in UMLS.

Consider now the coverage of the two dictionaries on the annotated data used for training/testing the decision tree learner: of the modifier nouns (e.g., *age* in *age distribution*) attested in the data, 84.73% are found in UMLS and 88.70% are found in WordNet. This means that for some candidate nominalisations for which no argument-verb relations are found in our corpus, smoothing will not be possible if they are not listed in UMLS or WordNet. Given that WordNet has a better coverage than UMLS on our data set, it follows that the former has a better chance at recreating the missing argument frequencies.

An obvious question is whether UMLS and WordNet are complementary, i.e., do the nouns listed in both dictionaries overlap or not? A total of 19,364 argument

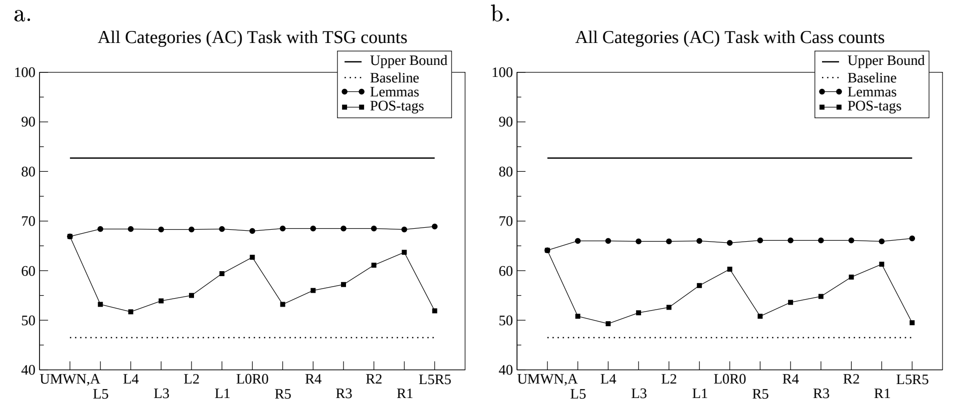

Fig. 8. Disambiguation Accuracy for Cass and TSG using smoothed counts and contextual features

nouns from the tuples extracted by the TSG are listed in both UMLS and WordNet. Of these, 33.06% are common in both dictionaries (i.e., 66.94% are listed either in UMLS or in WordNet but not in both). A similar picture emerges for Cass: a total 22,296 argument nouns are found in both dictionaries; of these, only 29.80% are common between UMLS and WordNet. This explains why the combination of UMLS and WordNet (see feature UMWN,A in Figure 7) outperforms all other features: it has greater smoothing power as it recreates missing argument frequencies by taking more data into account. The combination of the two dictionaries also has a better coverage on the annotated data, where 94.94% of the modifier nouns have a dictionary entry over UMLS' coverage of 84.73% and WordNet's coverage of 88.70%.

We next examine how accuracy is affected when contextual features are combined with features representing argument relations. We only display some (i.e., the most informative) of the feature combinations we examined. Furthermore, we focus on smoothed counts rather than raw frequencies as they deliver better accuracies on the classification task. For both parsers frequency counts are smoothed using the combination of UMLS and WordNet (UMWN); nominalization affixes (A) are used as an additional feature. Figures 8 and 9 display how the decision tree learner performs for the AC and OA task, respectively, when context is represented by lemmas or parts of speech. The letters L and R indicate whether left or right context is taken into account, wheres the numbers represent the size of the context (see x-axis). For comparison, we also include the learner's accuracy for the best non-contextual features (i.e., UMWN,A). So, L5 describes five lemmas to the left of the nominalisation target combined with UMWN,A; L0R0 denotes the absence of context; in this case only the head and modifier of the nominalisation are used as features together with UMWN,A.

As can be seen in Figure 8 part of speech tags yield lower performances than lemmas on the AC task. In fact, encoding context as parts of speech yields lower accuracies than using no context at all (see UMWN,A). A slight increase in accuracy over just using UMWN,A is obtained when context is encoded as lemmas. The highest accuracy is obtained with L5R5 for both TSG (68.4%) and Cass (64.1%). However

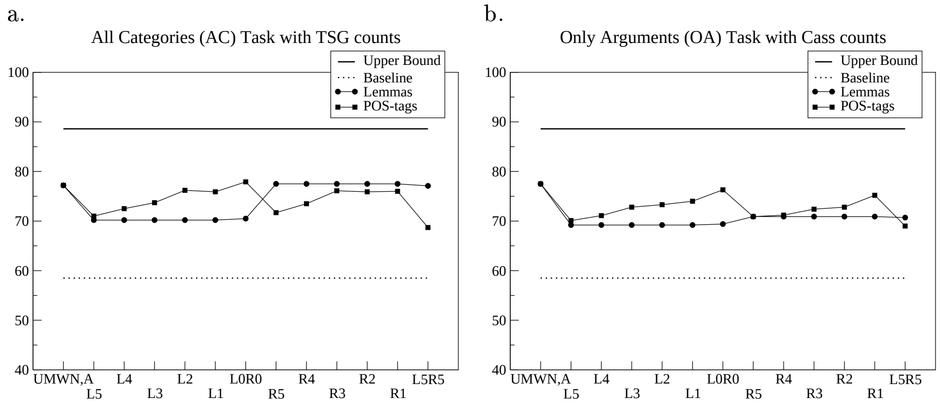

Fig. 9. Disambiguation Accuracy for Cass and TSG using smoothed counts and contextual features

the difference between L5R5 and UMWN,A is not significant, neither for Cass nor for the TSG (see Table 12 in the Appendix). Furthermore, there is no significant difference in the two parsers' performance (see Table 12). The lower performance of part of speech tags is not entirely unexpected: lemmas capture lexical dependencies which are somewhat lost when a more general level of representation is introduced.

A different picture emerges for the OA task (see Figure 9). Here, part of speech tags seem to outperform lemmas. This is particularly the case for the TSG when left context or no context is employed (see L5–L0R0 in Figure 9a). For Cass, part of speech tags perform better than lemmas for left context and for narrow right contexts (see L5–L0R0 and R3–R1 in Figure 9b). Recall that the OA task excludes NAs and NVs, i.e., non-argument classes. It is not therefore surprising that part of speech tags become more relevant in the OA task as they potentially express syntactic differences that characterize different argument relations. However, the contribution of context does not seem to enhance classification accuracy. Both for Cass and the TSG the best accuracy is achieved with no context (L0R0). The latter is not significantly different from UMWN,A (see Table 13 in the Appendix).

In sum, our results demonstrate that the best accuracies on the interpretation task are achieved when smoothed frequency counts are combined with information about the nominalisation affixes. Inclusion of contextual features does not yield a better performance. Our results further indicate that the underlying parser has a small effect on the interpretation task (see Tables 6 and 7). We further show that the type and size of the taxonomy used for recreating the argument frequencies has an impact on classification accuracy. Our best results were obtained when UMLS was combined with WordNet. Finally, our accuracies are expectedly lower than the ones reported in (Lapata, 2002) who only attempts a binary classification of nominalisations (SUBJ, OBJ) for domain independent text and achieves a performance of 80%. We obtain a reasonable performance for biomedical text by using generic NLP resources. We achieve an accuracy of 68.9% on the AC task (using TSG and features UMWN,A and L5R5) and an accuracy of 77.5% on the OA task (using TSG and features UMWN,A and L0R0). We also show that the use of the XML-based LT TTT toolset provides a highly flexible pipeline architecture that encourages reusable and modular processing. A major strength of this approach is the way in which knowledge of the text, represented as XML annotations, is incrementally computed by the use of successive modules. These pre-processing techniques are critical for the preparation of highly complex data for higher level parsing tasks.

7 Conclusions

In this paper we have investigated the use of generic, state-of-the-art NLP tools for performing various NLP semantic tasks over real corpus data in the biomedical domain. We focussed on two complementary tasks: generating a logical form for the sentences via 'deep' parsing with a hand-crafted grammar; and interpreting compound nouns with deverbal heads. The former task is challenging given that technical terms were pervasive in the corpus and our grammar lacked lexical coverage for a large portion of its sentences (approximately 98% in our case). The latter task is challenging because the semantic relation in a compound between a modifier and its head is linguistically implicit; indeed, this is an interpretation task that lies at the semantics/pragmatics interface (Hobbs *et al.*, 1993). Furthermore, performance on both of these tasks are adversely affected by the abundance of 'inherent messiness' in the data; e.g., equations and units of measurement which can create misleading amounts of syntactic complexity, making it harder to process medical data and discover novel semantic relations.

There are a number of lessons that we can learn from the experiments we have reported here. First, pre-processing the corpus data was crucial to success. Preprocessing allowed us to (a) obtain accurate information about words (their part-of-speech and their stems); and (b) 'package up' expressions such as units of measurement so that they were treated as unanalysable by the parser. Performing these tasks prior to parsing dramatically improved parse coverage with a hand-crafted grammar. The flexibility and robustness of the XML-tools was essential to the success of these pre-processing stages.

Second, through exploiting meaning paraphrases and surface cues, it is possible to acquire a reliable model of compound noun interpretation, thanks to the accuracy and robustness of current state-of-the-art broad coverage parsers such as Cass and the TSG parser. Overall, the models of nominalisation are more accurate when lexical taxonomies are used for smoothing and they take context into account. But there was no clear 'winner' between using Cass vs. the TSG parser. It seems that the trade-off between partial analyses vs. parse accuracy were not as important for this task as the presence vs. absence of smoothing and the size and coverage of the taxonomy used for the smoothing task. However, it remains for future work to investigate whether utilising a grammar as rich as the ANLT grammar, which would have to be combined with statistical parsing, would improve overall performance on this task. It may conceivably do this, since the ANLT grammar has access to the subcategorisation frames of all verbs, nouns and adjectives that appear in the Longmans Dictionary of Contemporary English (LDOCE), and consequently it may improve discrimination between complements vs. adjuncts compared with the other parsers we investigated, leading to more accurate predictions about grammatical relations. This is yet to be explored.

The resulting models of compound noun interpretation are highly complex, making it hard to tease apart the relative importance of the different parameters that we used in the various models. However, in light of the good performance of all of these models on a difficult interpretation task, we advocate the use of the 'knowledge poor' approach we developed here, relying on off-the-shelf NLP components and available taxonomies.

References

Abney, S. 1996. Partial parsing via finite-state cascades. *Pages 8–15 of:* Carroll, John (ed), *Workshop on Robust Parsing*. ESSLLI, Prague.

Andrade, MA, & Valencia, A 1998. Automatic extraction of keywords from scientific text: Application to the knowledge domain of protein families. *Bioinformatics*, 600–607.

Blaschke, C, Andrade, MA, Ouzounis, C, & Valencia, A. 1999. Automatic extraction of biological information from scientific text: Protein-protein interactions. *Pages 60–67 of: Proceedings of the 7th International Conference on Intelligent Systems for Mollecular Biology.*

Briscoe, EJ, & Carroll, J 2002. Robust accurate statistical annotation of general text. *Pages 1499–1504 of: Proceedings of the 3rd International Conference on Language Resources and Evaluation.* Las Palmas, Gran Canaria.

Briscoe, EJ, & Carroll, J. 1993. Generalised Probabilistic LR Parsing of Natural Language Corpora with Unification Grammars. *Computational linguistics*, **19**(1), 25–60.

Burnard, L. 1995. *Users Guide for the British National Corpus.* British National Corpus Consortium, Oxford University Computing Service.

Carroll, J, & Briscoe, EJ. 2002. High precision extraction of grammatical relations. *Pages 134–140 of: Proceedings of the 19th International Conference on Computational Linguistics (COLING).* Taipei, Taiwan.

Carroll, J, & Grover, C. 1988. The Derivation of a Large Computational Lexicon of English from LDOCE. *In:* Boguraev, B, & Briscoe, EJ (eds), *Computational Lexicography for Natural Language Processing.* Longman, London.

Carroll, J, Briscoe, EJ, & Sanfilippo, A. 1998. Parser Evaluation: A Survey and a New Proposal. *Pages 447–454 of: In Proceedings of the 1st International Conference on Language Resources and Evaluation.*

Carroll, J, Minnen, G, & Briscoe, EJ. 1999. Corpus annotation for parser evaluation. *Pages 35–41 of:* Uszkoreit, Hans, Brants, Thorsten, & Krenn, Brigitte (eds), *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora.*

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20**, 37–46.

Cracen, M, & Kumlien, J. 1999. Constructing biological knowledge bases by extracting information from text sources. *Pages 77–86 of: In Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology.*

Cunningham, H, Maynard, D, Bontscheva, K, & Tablan, C. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Pages 168–175 of: In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics.* Philadelphia, U.S.A.

Dagan, I, Lee, L, & Pereira, FCN 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, **34**(1–3), 43–69.

Elworthy, D. 1994. Does Baum-Welch re-estimation help taggers? *Pages 53–58 of: Proceedings of the 4th Conference on Applied Natural Language Processing.*

FININ, T. 1980. The semantic interpretation of nominal compounds. *Pages 310–315 of: Proceedings of 1st National Conference on Artificial Intelligence.*

FUKUDA, K, TAMURA, A, TSUNODA, T, & TAKAGI, T. 1998. Toward information extraction: Identifying protein names from biological papers. *Pages 707–718 of: Proceedings of the 3rd Annual Pacific Symposium on Biocomputing.*

GARSIDE, R. 1987. The CLAWS word-tagging system. *In:* GARSIDE, ROGER, LEECH, GEOFFREY, & SAMPSON, GEOFFREY (eds), *The Computational Analysis of English.* Longman, London.

GAZDAR, G, KLEIN, E, PULLUM, G, & SAG, I. 1985. *Generalized Phrase Structure Grammar.* London: Basil Blackwell.

GROVER, C, & LASCARIDES, A. 2001. XML-based data preparation for robust deep parsing. *In: Proceedings of the Joint EACL-ACL Meeting (ACL-EACL 2001).*

GROVER, C, MATHESON, C, MIKHEEV, A, & MOENS, M. 2000. LT TTT—a flexible tokenisation tool. *Pages 1147–1154 of: LREC 2000—Proceedings of the Second International Conference on Language Resources and Evaluation.*

HERSH, W, BUCKLEY, C, LEONE, TJ, & HICKAM, D 1994. OHSUMED: an interactive retrieval evaluation and new large test collection for research. *Pages 192–201 of:* CROFT, W. BRUCE, & VAN RIJSBERGEN, C. J. (eds), *Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval.*

HOBBS, JR, STICKEL, M, APPELT, D, & MARTIN, P 1993. Interpretation as abduction. *Artificial Intelligence*, **63**(1–2), 69–142.

HUMPHREYS, K, DEMETRIOU, G, & GAIZAUSKAS, R. 2000. Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. *Pages 505–516 of: In Proceedings of the 5th Annual Pacific Symposium on Biocomputing.*

HUMPHREYS, L, LINDBERG, DAB, SCHOOLMAN, HM, & BARNETT, GO 1998. The unified medical language system: An informatics research collaboration. *Journal of the American Medical Informatics Association*, **1**(5), 1–13.

IDE, N, BONHOMME, P, & ROMARY, L. 2000. XCES: An XML-based Encoding Standard for Linguistic Corpora. *In Proceedings of the Second International Language Resources and Evaluation Conference.*

ILIOPOULOS, I, ENRIGHT, A J, & OUZOUNIS, C. 2001. Textquest: Document clustering of medline abstracts for concept discovery in molecular biology. *Pages 384–395 of: Proceedings of the 6th Annual Pacific Symposium on Biocomputing.*

ISABELLE, P. 1984. Another look at nominal compounds. *Pages 509–516 of: Proceedings of the 10th International Conference on Computational Linguistics.*

JONES, B. 1995. Predicating nominal compounds. *Pages 130–135 of: Proceedings of 17th Annual Conference of the Cognitive Science Society.*

LAPATA, M. 2002. The disambiguation of nominalisations. *Computational Linguistics*, **28**(3).

LAPATA, M, KELLER, F, & MCDONALD, S. 2001. Evaluating smoothing algorithms against plausibility judgments. *Pages 346–353 of: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics.*

LAUER, M. 1995. *Designing Statistical Language Learners: Experiments on Compound Nouns.* Ph.D. thesis, Macquarie University.

LEONARD, R. 1984. *The Interpretation of English Noun Sequences on the Computer.* Amsterdam: North-Holland.

LEVI, JN 1978. *The Syntax and Semantics of Complex Nominals.* New York: Academic Press.

MACLEOD, C, GRISHMAN, R, MEYERS, A, BARRETT, L, & REEVES, R. 1998. Nomlex: A lexicon of nominalizations. *Pages 187–193 of: Proceedings of the 8th International Congress of the European Association for Lexicography.*

Marcus, MP, Santorini, B, & Marcinkiewicz, M. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, **19**(2), 313–330.

Marsh, E. 1984. A computational analysis of complex noun phrases in Navy messages. *Pages 505–508 of: Proceedings of the 10th International Conference on Computational Linguistics*.

McDonald, D. 1982. *Understanding Noun Compounds*. Ph.D. thesis, Carnegie Mellon University.

McKelvie, D. 1999. *XMLPERL 1.0.4.: XML Processing Software*.

Mikheev, A. 1997. Automatic rule induction for unknown word guessing. *Computational Linguistics*, **23**, 405–423.

Miller, GA, Beckwith, R, Fellbaum, C, Gross, D, & Miller, Katherine J. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, **3**(4), 235–244.

Minnen, G, Carroll, J, & Pearce, D. 2000. Robust, applied morphological generation. *Pages 716–721 of: Proceedings of 1st International Natural Language Generation Conference*.

Proux, D, Rechenmann, F, & Julliard, L. 2000. A pragmatic information extraction strategy for gathering data on genetic interactions. *Pages 279–285 of: In Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*.

Pustejovsky, J, no, J Casta Cohran, B, Kotecki, M, & Morrell, M. 2001. Automatic extraction of acronym-meaning pairs from medline databases. *Pages 371–375 of:* Patel, V L, Rogers, R, & Haux, R (eds), *In Proceedings of the 10th World Congress on Medical Informatics*. London: Amsterdam: IOS Press.

Pustejovsky, J, no, J Casta Cochran, B, & Kotecki, M. 2002. Robust relational parsing over biomedical literature: Extracting inhibit relations. *In: In Proceedings of the 7th Annual Pacific Symposium on Biocomputing*.

Quinlan, RJ 1993. *C4.5: Programs for Machine Learning*. Series in Machine Learning. San Mateo, CA: Morgan Kaufman.

Quirk, R, Greenbaum, S, Leech, G, & Svartvik, J. 1985. *A Comprehensive Grammar of the English Language*. London: Longman.

Rappaport, M, & Levin, B 1990. *-er* nominals: Implications for the theory of argument structure. *In:* Wehrli, E., & Stowell, T. (eds), *Syntax and Semantics volume 26: Syntax and the Lexicon*. Academic Press.

Resnik, PS. 1993. *Selection and Information: A Class-based Approach to Lexical Relationships*. Ph.D. thesis, University of Pennsylvania.

Rindfleisch, CT, Rayan, JV, & Lawrence, H. 2000. Extracting molecular binding relationships from biomedical text. *Pages 188–195 of: Proceedings of the 6th Applied Natural Language Conference and the 1st North American Annual Meeting of the Association for Computational Linguistics*.

Rosario, B, & Hearst, M. 2001. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. *Pages 82–90 of: Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing*.

Sekimizu, T, Park, HS, & Tsujii, J. 1998. Identifying the interaction between genes and gene products based on frequently seen verbs in medline abstracts. *Pages 62–71 of:* Miyano, S, & Takagi, T (eds), *In Proceedings of Genome Informatics 1998*. Yebisu Garden Place, Tokyo: Universal Academy Press.

Thompson, HS, Tobin, R, McKelvie, D, & Brew, C. 1997. *LT XML: Software API and toolkit for XML processing*.

Vanderwende, L. 1994. Algorithm for automatic interpretation of noun sequences. *Pages 782–788 of: Proceedings of the 15th International Conference on Computational Linguistics*.

Warren, B. 1978. *Semantic Patterns of Noun-Noun Compounds*. Göteborg: Acta Universitatis Gothoburgensis.

WITTEN, IH, & FRANK, E. 2000. *Data mining: Practical Machine Learning Tools and Techniques with Java Implementations.* San Francisco, CA: Morgan Kaufman.

YAKUSHIJI A, YUKA TATEISI, MIYAO YUSUKE & JUN'ICHI TSUJII. 2001. *Event extraction from biomedical papers using a full parser. Pages 408–419 of: Proceedings of the 6th Pacific Symposium on Biocomputing.* Hawaii, U.S.A.

Appendix

Using a χ^2 test we examined whether the accuracies shown in Figures 7a and 7b differ significantly. The χ^2 values and significance levels are shown below. Tables 8 and 9 correspond to Figure 7a, whereas Tables 10 and 11 correspond to Figure 7b. In Tables 12 and 13 we examine whether the best contextual features yield accuracies significantly different from using only UMWN,A.

Table 8. *Significance testing,* TSG *parser, OA task*

	B	F	F,A	WN	WN,A	UMLS	UMLS,A	UMWN,A
F	11.28**							
F,A	26.49**	3.23						
WN	43.06**	10.37**	2.03					
WN,A	59.05**	18.94**	6.56*	1.29				
UMLS	17.36**	0.66	0.97	5.81*	12.56**			
UMLS,A	34.51**	6.39*	0.54	0.48	3.35	2.95		
UMWN,A	93.13**	40.20**	20.77**	9.84**	4.01*	30.65**	14.65**	
UB	315.97**	214.44**	167.59**	134.26**	110.19**	192.64**	150.0**	73.29**

Table 9. *Significance testing, Cass, OA task*

	B	F	F,A	WN	WN,A	UMLS	UMLS,A	UMWN,A
F	9.37**							
F,A	26.05**	4.21*						
WN	24.34**	3.53	0.03					
WN,A	37.59**	9.51*	1.07	1.46				
UMLS	23.92**	3.37	0.05	0.00	1.56			
UMLS,A	23.50**	3.22	0.07	0.01	1.67	0.00		
UMWN,A	69.46**	28.15**	10.65**	11.79**	4.97*	12.09**	12.39**	
UB	317.37**	223.92**	169.71**	173.99**	145.03**	175.06**	176.14*	98.06**

*$p < .05$ (2-tailed) **$p < .01$ (2-tailed)

Table 10. *Significance testing,* TSG*parser, AC task*

	B	F	F,A	WN	WN,A	UMLS	UMLS,A	UMWN,A
F	7.56**							
F,A	23.56**	4.46*						
WN	45.43**	16.31**	3.63					
WN,A	53.33**	20.99**	6.13*	0.33				
UMLS	26.28**	5.70*	0.08	2.66	4.85*			
UMLS,A	33.26**	9.19**	0.85	0.97	2.42	0.42	1.19	
UMWN,A	88.64**	32.58**	20.61**	6.24*	3.89*	30.58**	19.79**	
UB	257.06**	157.39**	130.91**	90.26**	80.97**	153.20**	128.95**	50.44**

Table 11. *Significance testing, Cass, AC task*

	B	F	F,A	WN	WN,A	UMLS	UMLS,A	UMWN,A
F	7.56**							
F,A	23.56**	4.46*						
WN	45.43**	16.10**	3.63					
WN,A	53.33**	20.99**	6.13*	0.33				
UMLS	26.28**	0.08	0.08	2.66	4.85*			
UMLS,A	33.26**	0.85	0.85	0.97	2.42	0.42	0.42	
UMWN,A	91.43**	47.19**	22.86**	8.33**	5.36*	20.33**	14.95**	
UB	257.06**	182.26**	132.88**	94.60**	84.31**	126.99**	113.63**	48.37**

*$p < .05$ (2-tailed) **$p < .01$ (2-tailed)

Table 12. *Significance testing, AC task*

	B	UMWN,A$_{TSG}$	UMWN,A$_{Cass}$	L5R5$_{TSG}$	L5R5$_{Cass}$
UMWN,A$_{TSG}$	93.93**				
UMWN,A$_{Cass}$	69.46**	1.92			
L5R5$_{TSG}$	90.55**	0.03	5.69*		
L5R5$_{Cass}$	90.55**	0.03	1.45	1.40	
UB	317.37**	73.29**	76.32**	113.77**	98.06**

*$p < .05$ (2-tailed) **$p < .01$ (2-tailed)

Table 13. *Significance testing, OA task*

	B	UMWN,A$_{\text{TSG}}$	UMWN,A$_{\text{Cass}}$	L5R5$_{\text{TSG}}$	L5R5$_{\text{Cass}}$
UMWN,A$_{\text{TSG}}$	88.64**				
UMWN,A$_{\text{Cass}}$	91.43**	0.02			
L5R5$_{\text{TSG}}$	76.29**	0.49	0.73		
L5R5$_{\text{Cass}}$	79.71**	0.25	0.43	0.73	
UB	257.06**	50.44**	48.37**	60.52**	57.57**

*$p < .05$ (2-tailed) **$p < .01$ (2-tailed)