3

Comparative Evaluation of Grammatical Annotation Models

Eric Steven Atwell [1]

University of Leeds

3.1 Introduction

The objective of the IPSM Workshop was to empirically evaluate a number of robust parsers of English, in essence by giving each parser a common test-set of sentences, and counting how many of these sentences each parser could parse correctly. Unfortunately, what counts as a 'correct' parse is different for each parser, as the output of each is very different in both format and content: they each assume a different grammar model or parsing scheme for English. This chapter explores these differences in parsing schemes, and discusses how these differences should be taken into account in comparative evaluation of parsers. Chapter 2 suggests that one way to compare parser outputs is to convert them to a dependency structure. Others, e.g. (Atwell 1988), (Black et al 1993) have advocated mapping parses onto simple context-free constituency structure trees. Unfortunately, in mapping some parsing schemes onto this kind of 'lowest common factor', a lot of syntactic information is lost; this information is vital to some applications.

[1] Address: Centre for Computer Analysis of Language And Speech (CCALAS), Artificial Intelligence Division, School of Computer Studies, The University of Leeds, LEEDS LS2 9JT, Yorkshire, England. Tel: +44 113 2335761, Fax: +44 113 2335468, Email: eric@scs.leeds.ac.uk, WWW: http://agora.leeds.ac.uk/ccalas/

I gratefully acknowledge the UK Engineering and Physical Sciences Research Council (EPSRC) for funding the AMALGAM project; the UK Higher Education Funding Councils' Joint Information Systems Committee New Technologies Initiative (HEFCs' JISC NTI) for funding the NTI-KBS/CALAS project and my participation in the IPSM Workshop, and the EU for funding my participation in the 1996 EAGLES Text Corpora Working Group Workshop. I also gratefully acknowledge the contributions of co-researchers on the AMALGAM project, John Hughes and Clive Souter, and the various contributors to the AMALGAM MultiTreebank including John Carroll, Alex Fang, Geoffrey Leech, Nelleke Oostdijk, Geoffrey Sampson, Tim Willis, and (last but not least!) all the contributors to this book

The differences between parsing schemes is a central issue in the project AMALGAM: Automatic Mapping Among Lexico-Grammatical Annotation Models. The AMALGAM project at Leeds University is investigating the problem of comparative assessment of rival syntactic analysis schemes. The focus of research is the variety of lexico-grammatical annotation models used in syntactically-analysed Corpora, principally those distributed by ICAME, the International Computer Archive of Modern English based at Bergen University. For more details, see (Atwell et al, 1994a,b), (Hughes & Atwell, 1994), (Hughes et al, 1995), (Atwell 1996), (AMALGAM 1996), (ICAME 1996).

Standardisation of parsing schemes is also an issue for the European Union-funded project EAGLES: Expert Advisory Group on Language Engineering Standards (EAGLES 1996). Particularly relevant is the 'Final Report and Guidelines for the Syntactic Annotation of Corpora' (Leech et al, 1995); [2] this proposes several layers of recommended and optional annotations, in a hierarchy of importance.

3.2 Diversity in Grammars

The parsers in this book are diverse, in that they use very different algorithms to find parse-trees. However, to a linguist, the differences in underlying grammars or parsing schemes are more important. The differences are not simply matters of representation or notation (although these alone cause significant problems in evaluation, eg in alignment). A crucial notion is *delicacy* or level of detail in grammatical classification. This chapter explores some possible metrics of delicacy, applied to comparative evaluation of the parsing schemes used in this book.

Delicacy of parsing scheme clearly impinges on the accuracy of a parser. A simple evaluation metric used for parsers in this book is to count how often the parse-tree found is 'correct', or how often the 'correct' parse-tree is among the set or forest of trees found by the parser. However, this metric is unfairly biased against more sophisticated grammars, which attempt to capture more fine-grained grammatical distinctions. On the other hand, this metric would favour an approach to syntax modelling which lacks this delicacy. Arguably it is not sensible to seek a scale of accuracy applicable across all applications, as different applications require different levels of parsing; see, for example, (Souter & Atwell 1994). For some applications, a skeletal parser is sufficient, so we can demand high accuracy: for example, n-gram grammar modelling for speech or script recognition systems (see next section); parsing

[2] DISCLAIMER: My description of the EAGLES guidelines for the syntactic annotation of corpora is based on the PRE-RELEASE FINAL DRAFT version of this Report, dated July 31st 1995; the final version, due for publication in 1996, may include some changes.

corpus texts prior to input to a lexicographer's KWIC workbench; or error-detection in Word Processor text. For these applications, parsing is simply an extra factor or guide towards an improved 'hit rate' - all *could* still work without syntactic analysis and annotation, but perform better with it. Other applications require detailed syntactic analysis, and *cannot* function without this; for example, SOME (but by no means all!) NLP systems assume that the parse-tree is to be passed on to a semantic component for knowledge extraction, so need richer syntactic annotation.

3.3 An Extreme Case: the 'Perfect Parser' from Speech Recognition

The variability of delicacy is exemplified by one approach to parsing which is widely used in Speech And Language Technology (SALT). Most large-vocabulary English speech recognition systems use a word N-gram language model of English grammar: syntactic knowledge is captured in a large table of word bigrams (pairs), trigrams (triples), ... N-grams (see surveys of large-vocabulary speech recognition systems, eg (HLTsurvey 1995), (comp.speech 1996). This table is extracted or *learnt* from a training corpus, a representative set of texts in the domain of the speech recogniser; training involves making a record of every N-gram which appears in the training text, along with its frequency (eg in this Chapter the bigram *recognition systems* occurs 4 times). The 'grammar' does not make use of phrase-structure boundaries, or even word-classes such as Noun or Verb. The job of the 'parser' is not to compute a parse-tree for an input sentence, but to estimate a syntactic probability for the input word-sequence. The 'parser' is guaranteed to come up with SOME analysis (ie syntactic probability estimate) for ANY input sentence; in this sense it is a 'perfect' parser, outperforming all the other parsers in this book.

However, this sort of 'parsing' is inappropriate for many IPSM applications, where the assumption is that some sort of parse-tree is to be passed on to a semantic component for knowledge extraction. In linguistic terms, the Speech Recognition grammar model has insufficient delicacy (or no delicacy at all!)

3.4 The Corpus as Empirical Definition of Parsing Scheme

A major problem in comparative evaluation of parsing schemes is pinning down the DEFINITIONS of the parsing schemes in question. Generally

the parser is a computer program which can at least in theory be directly examined and tested; we can evaluate the algorithm as well as the output. Parsing schemes tend to be more intangible and ephemeral: generally the parsing scheme exists principally in the mind of the expert human linguist, who decides on issues of delicacy and correctness of parser output. For most of the syntactically-analysed corpora covered by the AMALGAM project, we have some 'manual annotation handbook' with general notes for guidance on definitions of categories; but these are not rigorously formal or definitive, nor are they all to the same standard or level of detail. For the AMALGAM project, we were forced to the pragmatic decision to accept the tagged/parsed Corpus itself as definitive of the tagging/parsing scheme for that Corpus. For example, for Tagged LOB, (Johansson et al, 1986) constitutes a detailed Manual, but for the SEC parsing scheme we have to rely on a list of categories and some examples of how to apply them; so we took the LOB and SEC annotated corpora themselves as definitive examples of respective syntactic analysis schemes.

Another reason for relying on the example data rather than explanatory manuals is the limitation of the human mind. Each lexicogrammatical annotation model for English is so complex that it takes an expert human linguist a long time, months or even years, to master it. For example, (Sampson, 1995), the definition of the SUSANNE parsing scheme, is over 500 pages long. To compare a variety of parsing schemes via such manuals, I would have to read, digest and comprehensively cross-reference several such tomes. Perhaps a couple of dozen linguists in the world could realistically claim to be experts in two rival Corpus parsing schemes, but I know of none who are masters of several. I have been forced to the conclusion that it is unreasonable to ask anyone to take on such a task (and I am not about to volunteer myself!)

This pragmatic approach is also necessary with the parsing schemes used in this book. Not all the parsing schemes in use have detailed definition handbooks, as far as I am aware; at the very least, I do not have access to all of them. So, comparative evaluation of parsing schemes must be based on the small corpus of test parse-trees presented at the IPSM workshop. Admittedly this only constitutes a small sample of each parsing scheme, but hopefully the samples are comparable subsets of complete grammars, covering the same set of phrase-types for each parsing scheme. This should be sufficient to at least give a relative indicator of delicacy of parsing schemes.

3.5 Towards a MultiTreebank

One advantage of the IPSM exercise is that all parsers were given the same of sentences to parse, so we have directly-comparable parses for given sentences; the same is not true for ICAME parsed corpora, also called *treebanks*. Even if we assume that, for example, the SEC treebank embodies the definition of the SEC parsing scheme, the POW treebank defines the POW parsing scheme, etc, there is still a problem in comparing delicacy across parsing schemes. The texts parsed in each treebank are different, which complicates comparison. For any phrase-type or construct in the SEC parsing scheme, it is not straightforward to see its equivalent in POW: this involves trawling through the POW treebank for similar word-sequences. It would be much more straightforward to have a single text sample parsed according to all the different schemes under investigation, a MultiTreebank. This would allow for direct comparisons of rival parses of the same phrase or sentence. However, creation of such a resource is very difficult, requiring the cooperation and time of of the research teams responsible for each parsed corpus and/or robust parser.

A first step towards a prototype MultiTreebank was achieved in the Proceedings of the IPSM workshop, which contained the output of several parsers' attempts to parse half a dozen example sentences taken from software manuals. Unfortunately each sentence caused problems for one or more of the parsers, so this mini-MultiTreebank has a lot of 'holes' or gaps. As an example for further investigation, I selected one of the shortest sentences (hence, hopefully, most grammatically straightforward and uncontroversial), which most parsers had managed to parse:

```
Select the text you want to protect.
```

To the example parses produced by IPSM participants, I have been able to add parses conformant to the parsing schemes of several large-scale English treebanks, with the assistance of experts in several of these parsing schemes; see (AMALGAM 1996).

3.6 Vertical Strip Grammar: a Standard Representation for Parses

Before we can compare delicacy in the way two rival parsing-schemes annotate a sentence, we have to devise a parsing-scheme-neutral way of representing rival parse-trees, or at least of mapping between the schemes. I predict that most readers will be surprised by the wide diversity of notation used by the parsers taking part in the IPSM workshop; I certainly was. This can only confuse attempts to compare underlying grammatical classification distinctions.

This is a major problem for the AMALGAM project. Even Corpora which are merely wordtagged (without higher syntactic phrase boundaries marked) such as BNC, Brown etc, are formatted in a bewildering variety of ways. As a 'lowest common factor' , or rather, a 'lowest common anchor-point', each corpus could be visualised as a sequence of *word + wordtag* pairs. Even this simplification raises problems of incompatible alignment and segmentation. Some lexico-grammatical annotation schemes treat various idiomatic phrases, proper-name-sequences, etc as a single token or 'word'; whereas others split these into a sequence of words to be assigned separate tags. Some parsing schemes split off certain affixes as separate lexemes or tokens requiring separate tags; while others insist that a 'word' is any character-sequence delimited by spaces or punctuation.

However, putting this tokenisation problem to one side, it is useful to model any wordtagged Corpus as a simple sequence of *word + wordtag* pairs. This can be used to build N-gram models of tag-combination syntax. For full parses, the words in the sentence still constitute a 'lowest common anchor point', so we have considered N-gram-like models of parse-structures. For example, take the EAGLES basic parse-tree:

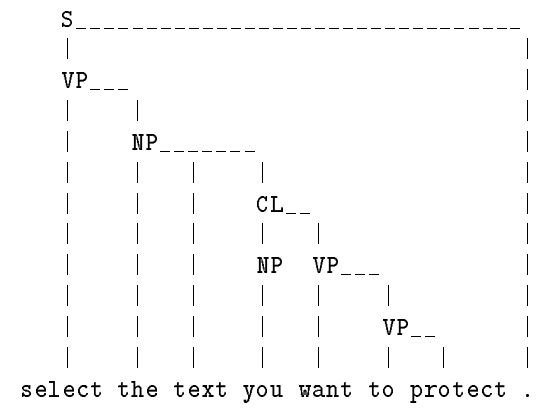
```
[S[VP select [NP the text [CL[NP you NP][VP want [VP to
 protect VP]VP]CL]NP]VP] . S]
```

Words are 'anchors', with *hypertags* between then showing opening and/or closing phrase boundaries. These hypertags are inter-word grammatical tokens alternating with the words, with a special NULL hypertag to represent absence of inter-word phrase boundary:

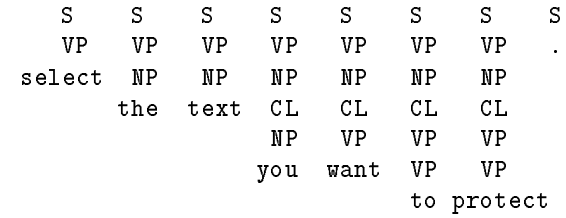
```
        [S[VP
select
        [NP
the
        NULL
text
        [CL[NP
you
        NP][VP
want
        [VP
to
        NULL
protect
        VP]VP]CL]NP]VP]
.
        S]
```

When comparing rival parses for the same sentence, we can 'cancel out' the words as a common factor, leaving only the grammatical information assigned according to the parsing scheme. So, one way to normalise parse-structures would be to represent them as an alternating sequence of wordtags and inter-word structural information; this would render transparent the amount and delicacy of structural classificatory information. This would allow us to try quantitative comparison metrics, eg the length of the hypertag-string.

However, this way of building an N-gram like model is heavily reliant on phrase structure bracketting information, and so is not appropriate for some IPSM parsing schemes, those with few or no explicit phrase boundaries. The problem is that all the parses do have WORDS in common, but not all have inter-word bracketting information. An N-gram-like model which has states for words (but not inter-word states) may be more general. A variant N-gram-like model which meets this requirement is a Vertical Strip Grammar, as used in the Vertical Strip Parser (O'Donoghue, 1993). In this, a parse-tree is represented as a series of Vertical Strips from root to leaves. For example, given the syntax tree:



This can be chopped into a series of Vertical Strips, one for each path from root S to each leaf:



This Vertical Strip representation is highly redundant, as the top of each strip shares its path from the root with its predecessor. So, the VSG representation only records the path to each leaf from the point of divergence from the previous Strip:

```
     S
     VP                                  .
  select  NP
          the  text  CL
                     NP   VP
                    you  want  VP
                               to protect
```

This VSG representation captures the grammatical information tied to each word, in a compact normalised form. Output from the various parsers can likewise be mapped onto an N-gram-like normalised VSG form:

```
Sentence:
 select   the   text   you   want   to   protect   .

ALICE:

 SENT                                  VP-INF
 AUX                  SENT             INF-MARK VP-INF
 ?       NP           NP      VP-ACT   to       protect
 select  DET   NOUN   you     want
            the   text

ENGCG:

 @+FMAINV @DN> @OBJ @SUBJ @+FMAINV @INFMARK> @-FMAINV .
 V        DET  N    PRON  V        INFMARK   V
 select   the  text you   want     to        protect  .
```

The ENGCG output is unusual in that it provides very detailed word-category labelling for each word, but only minimal structural information. In the above I have omitted the wordclass subcategory information, eg

```
select: <*> <SVO> <SV> <P/for> V IMP VFIN
```

LINK:

O			B			
W	D		C	S	TO	I
v	the	n	you	want	to	v
select		text				protect

PRINCIPAR:

VP						
Vbar						
V						
V_NP						
V_NP	NP					
select	Det	Nbar				
	the	N	CP			
		text	Op[1]			
			Cbar			
			IP			
			NP	Ibar		
			Nbar	VP		
			N	Vbar		
			you	V		
				V_CP		
				V_CP	CP	
				want	Cbar	
					IP	
					PRO	
					Ibar	
					Aux	VP
					to	Vbar
						V
						V_NP
						V_NP
						protect
						t[1]

PLAIN:

```
 ILLOC
 command
 PROPOS
 *       DIR_OBJ1
 imperat DETER   *       ATTR_ANY
 select  definit singula rel_clause
                 text    PRED
                         SUBJECT *       DIR_OBJ2
                         you     present clause
                                 want    PROPOS
                                         to      protect
```

RANLT:

```
 VP/NP
 select  N2+/DET
         the     N2-
                 N1/INFM
                 N1/RELM                 VP/TO
                 N1/N    S/THATL         to      VP/NP
                 text    S1a                     protect
                         N2+/PRO VP/NP           TRACE1
                         you     want            E
                                 TRACE1
                                 E
```

SEXTANT:

```
 VP      NP              NP      VP                      --
 INF     3       *       *       INF     TO      4       .
 select  DET     1       PRON    want    to      SUBJ    .
         DET     DOBJ    you                     INF
         the     NOUN                            protect
                 text
```

TINGYIN:

```
 8       3       1       5       3       7       5       0
 VB      DT      NN      PP      VBP     TO      VB      .
 select  the     text    you     want    to      protect .
```

TOSCA:

Unfortunately this was one of only a couple of IPSM test sentences that the TOSCA parser could not parse, due to the syntactic phenomenon known as 'raising': according to the TOSCA grammar, both the verbs 'select' and 'protect' require an object, and although in some deep sense 'the text' is the object of both, the TOSCA grammar does not allow for this construct. However, the TOSCA research team have kindly constructed a 'correct' parse for our example sentence, to compare with others, by parsing a similar sentence and then 'hand-editing' the similar parse-tree. This includes very detailed subclassification information with each label (see subsection 3.7.5, which includes the TOSCA 'correct' parse-tree). For my VSG normalisation I have ommitted this:

```
NOFU,TXTU
UTT,S                                                    PUNC,PM
V,VP   OD,NP                                             .
MVB,LV DT,DTP   NPHD,N NPPO,CL
Select DTCE,ART text   SU,NP   V,VP   OD,CL
       the             NPHD,PN MVB,LV TO,PRTC LV,VP
                       you     want   to      MVB,LV
                                              protect
```

3.7 EAGLES: A Multi-Layer Standard for Syntactic Annotation

This standard representation is still crude and appears unfair to some schemes, particularly dependency grammar which has no grammatical classes! Also, it assumes the parser produces a *single* correct parse-tree - is it fair to parsers (eg RANLT) which produce a *forest* of possible parses? It at least allows us to compare parser outputs more directly, and potentially to combine or merge syntactic information from different parsers.

Mapping onto a standard format allows us to focus on the substantive differences between parsing schemes. It turns out that delicacy is not a simple issue, as different parsers output very different kinds or levels of grammatical information. This brings us back to our earlier point: parsing schemes should be evaluated with respect to a given application, as different applications call for different levels of analysis.

To categorise these levels of grammatical analysis, we need a taxonomy of possible grammatical annotations. The European Commission-funded EAGLES project (Expert Advisory Group for Language Engineering Standards) has attempted to devise common standards for a range of NLP issues to cover the range of European Union languages.

The EAGLES draft Report on parsing schemes (Leech et al, 1995) suggests that these layers of annotation form a *hierarchy of importance*, summarised in Table 3.1 at the end of this Section.

The Report does not attempt formal definitions or stipulate standardised labels to be used for all these levels, but it does give some illustrative examples. From these I have attempted to construct the layers of analysis for our standard example sentence:

3.7.1 (a) Bracketing of segments

The Report advocates two formats for representing phrase structure, which it calls Horizontal Format and Vertical Format; see (Atwell, 1983).

In both, opening and closing phrase boundaries are shown by square brackets between words; in horizontal format the text reads horizontally down the page, one word per line, while in vertical format the text reads left-to-right across the page, interspersed with phrase boundary brackets:

```
[[ select [ the text [[ you ][ want [ to protect ]]]]] . ]
```

3.7.2 (b) Labelling of segments

This can also be represented compactly in vertical format:

```
[S[VP select [NP the text [CL[NP you NP][VP want [VP to
 protect VP]VP]CL]NP]VP] . S]
```

The EAGLES report recommends the use of the categories S (Sentence), CL (Clause), NP (Noun Phrase), VP (Verb Phrase), PP (Prepositional Phrase), ADVP (Adverb Phrase), ADJP (Adjective Phrase). Although the EAGLES standard does not stipulate any *obligatory* syntactic annotations, these phrase structure categories are *recommended*, while the remaining layers of annotation are *optional*. Thus the above EAGLES parse-tree can be viewed as a baseline 'lowest common factor' target for parsers to aim for.

3.7.3 (c) Showing dependency relations

The Report notes that: "as far as we know, the ENGCG parser is the only system of corpus annotation that uses dependency syntax", which makes the ENGCG analysis a candidate for the de-facto EAGLES standard for this layer. However, the dependency analysis is only partial - the symbol > denotes that a word's head follows, and only two such dependencies are indicated for our example sentence:

```
         >                          >
 select  the    text   you    want   to     protect .
```

The report cites three traditional ways of representing dependency analyses graphically; however, the first cited traditional method, using curved arrows drawn to link dependent words, is equivalent to the TINGYIN method using word-reference numbers:

```
8       3      1      5      3      7      5       0
1       2      3      4      5      6      7       8
select  the    text   you    want   to     protect .
```

3.7.4 (d) Indicating functional labels

The report cites SUSANNE, TOSCA and ENGCG as examples of parsing schemes which include syntactic function labels such as Subject, Object, Adjunct. In TOSCA output, every node-label is a pair of *Function,Category*; for example, *SU,NP* labels a Noun Phrase functioning as a SUbject. In the ENGCG analysis, function is marked by @:

```
@+FMAINV @D   @OBJ  @SUB  @+FMAINV @INFMARK @-FMAINV .
select   the  text  you   want     to       protect   .
```

3.7.5 (e) Marking subclassification of syntactic segments

Example subclassification features include marking a Noun Phrase as singular, or a verb Phrase as past tense. The TOSCA parser has one of the richest systems of subclassification, with several subcategory features attached to most nodes, lowercase features in brackets:

```
NOFU,TXTU()
 UTT,S(-su,act,imper,motr,pres,unm)
  V,VP(act,imper,motr,pres)
   MVB,LV(imper,motr,pres){Select}
  OD,NP()
   DT,DTP()
    DTCE,ART(def){the}
   NPHD,N(com,sing){text}
   NPPO,CL(+raisod,act,indic,motr,pres,unm,zrel)
    SU,NP()
     NPHD,PN(pers){you}
    V,VP(act,indic,motr,pres)
     MVB,LV(indic,motr,pres){want}
    OD,CL(-raisod,-su,act,indic,infin,motr,unm,zsub)
     TO,PRTCL(to){to}
     V,VP(act,indic,infin,motr)
      MVB,LV(indic,infin,motr){protect}
 PUNC,PM(per){.}
```

The ENGCG parsing scheme also includes subclassification features at the word-class level:

```
"select" <*> <SVO> <SV> <P/for> V IMP VFIN
"the" <Def> DET CENTRAL ART SG/PL
"text" N NOM SG
"you" <NonMod> PRON PERS NOM SG2/PL2
"want" <SVOC/A> <SVO> <SV> <P/for> V PRES -SG3 VFIN
"to" INFMARK>
"protect" <SVO> V INF
```

3.7.6 (f) Deep or 'logical' information

This includes *traces* or markers for extraposed or moved phrases, such as capturing the information that 'the text' is not just the Object of 'select' but also the (raised) Object of 'protect'. This is captured by the features *+raisod* and *-raisod* in the above TOSCA parse-tree; by cross-indexing of *Op[1]* and *t[1]* in the PRINCIPAR parse; and by *(TRACE1 E)* in the RANLT parse.

3.7.7 (g) Information about the rank of a syntactic unit

The Report suggests that "the concept of rank is applied to general categories of constituents, words being of lower rank than phrases, phrases being of lower rank than clauses, and clauses being of lower rank than sentences". This is not explicitly shown in most of the parser outputs, beyond the common convention that words are in lowercase while higher-rank units are in UPPERCASE or begin with an Uppercase letter. However, I believe that the underlying grammar models used in PRINCIPAR and RANLT do include a rank hierarchy of nominal units: NP-Nbar-N in PRINCIPAR, NP-N2-n1-N in RANLT.

3.7.8 (h) Special syntactic characteristics of spoken language

This layer includes special syntactic annotations for "a range of phenomena that do not normally occur in written language corpora, such as blends, false starts, reiterations, and filled pauses". As the IPSM test sentences were written rather than spoken texts, this layer does not apply to us. However, we have successfully applied the TOSCA and ENGCG parsers to spoken text transcripts at Leeds in the AMALGAM research project.

Layer	Explanation
(a)	Bracketing of segments
(b)	Labelling of segments
(c)	Showing dependency relations
(d)	Indicating functional labels
(e)	Marking subclassification of syntactic segments
(f)	Deep or 'logical' information
(g)	Information about the rank of a syntactic unit
(h)	Special syntactic characteristics of spoken language

Table 3.1: EAGLES layers of syntactic annotation, forming a hierarchy of importance.

Code	Explanation
A	Verbs recognised
B	Nouns recognised
C	Compounds recognised
D	Phrase Boundaries recognised
E	Predicate-Argument Relations identified
F	Prepositional Phrases attached
G	Coordination/Gapping analysed

Table 3.2: characteristics used in IPSM parser evaluation

3.7.9 Summary: a hierarchy of importance

Table 3.1 summarises the EAGLES layers of syntactic annotation, which form a hierarchy of importance. No parsing scheme includes all the layers a-g; different IPSM parsers annotate with different subsets of of the hierarchy.

3.8 Evaluating the IPSM Parsing Schemes against EAGLES

For the IPSM Workshop, each parsing scheme was evaluated in terms of "what kinds of structure the parser can in principle recognise". Each of the chapters after this one includes a table showing which of the characteristics in Table 3.2 are handled by the parser.

These characteristics are different from the layers of annotation in the EAGLES hierarchy, Table 3.1. They do not so much characterise

layer	a	b	c	d	e	f	g	score
ALICE	yes	yes	no	no	no	no	no	2
ENGCG	no	no	yes	yes	yes	no	no	3
LINK	no	no	yes	yes	no	no	no	2
PLAIN	yes	yes	no	yes	no	no	no	3
PRINCIPAR	yes	yes	yes	no	no	yes	yes	5
RANLT	yes	yes	no	no	no	yes	yes	4
SEXTANT	yes	yes	yes	yes	no	no	no	4
TINGYIN	no	no	yes	no	no	no	no	1
TOSCA	yes	yes	no	yes	yes	yes	no	5

Table 3.3: Summary Comparative Evaluation of IPSM Grammatical Annotation Models, in terms of EAGLES layers of syntactic annotation. Each cell in the table is labelled *yes* or *no* to indicate whether an IPSM parsing scheme includes an EAGLES layer (at least partially). *score* is a an indication of how many layers a parser covers.

the parsing scheme, but rather the degree to which the parser can apply it successfully. For example, criterion F does not ask whether the parsing scheme includes the notion of Prepositional Phrase (all except TINGYIN do, although only PRINCIPAR and TOSCA explicitly use the label PP); rather it asks whether the parser is 'in principle' able to recognise and attach Prepositional Phrases correctly. Furthermore, most of the characteristics relate to broad categories at the 'top' layers of the EAGLES hierarchy.

Table 3.3 is my alternative attempt to characterise the rival parsing schemes, in terms of EAGLES layers of syntactic annotation. Each IPSM parsing scheme is evaluated according to each EAGLES criterion; and each parsing scheme gets a very crude overall 'score' showing how many EAGLES layers are handled, at least partially.

Note that this based on my own analysis of output from the IPSM parsers, and I may have misunderstood some capabilities of the parsers. PRINCIPAR is unusual is being able to output two parses, to give both Dependency and Constituency analysis; I have included both in my analysis, hence its high 'score'. The TOSCA analysis is based on the 'handcrafted' parse supplied by the TOSCA team, given that their parser failed with the example sentence; I am not clear whether the automatic parser can label deep or 'logical' information such as the raised Object of *protect*.

3.9 Summary and Conclusions

In this chapter, I have attempted the comparative evaluation of IPSM grammatical annotation models or parsing schemes. The first problem is that the great variety of output formats hides the underlying substantive similarities and differences. Others have proposed mapping all parser outputs onto a Phrase-Structure tree notation, but this is arguably inappropriate to the IPSM evaluation exercise, for at least two reasons:

1. several of the parsers (ENGCG, LINK, TINGYIN) do not output traditional constituency structures, and
2. most of the parsers output other grammatical information which does not 'fit' and would be lost in a transformation to a simple phrase-structure tree.

The chapter by Lin proposes the alternative of mapping all parser outputs to a Dependency structure, but this is also inappropriate, for similar reasons:

1. most of the parsers do not output Dependency structures, so to force them into this minority representation would seem counter-intuitive; and
2. more importantly, *most* of the grammatical information output by the parsers would be lost in the transformation: dependency is only one of the layers of syntactic annotation identified by EAGLES.

In other words, mapping onto either constituency or dependency structure would constitute 'degrading' parser output to a lowest common factor, which is a particularly unfair evaluation procedure for parsers which produce 'delicate' analyses, covering several layers in the EAGLES hierarchy.

As an alternative, I have transformed IPSM parser outputs for a simple example sentence onto a compromise Vertical Strip Grammar format, which captures the grammatical information tied to each word, in a compact normalised form. The VSG format is derived from a constituent-structure tree, but it can accommodate partial structural information output by ENGCG and LINK parsers. The VSG format is NOT intended for use in automatic parser evaluation experiments, as clearly the VSG forms of rival parser outputs are still clearly different, not straightforwardly comparable. The VSG format is intended as a tool to enable linguists to compare grammatical annotation models, by factoring out notational from substantive differences.

The EAGLES report on European standards for syntactic annotation identifies a hierarchy of levels of annotation. Transforming IPSM parser

layer	a	b	c	d	e	f	g	score
ALICE	7	6	0	0	0	0	0	13
ENGCG	0	0	5	4	3	0	0	12
LINK	0	0	5	4	0	0	0	9
PLAIN	7	6	0	4	0	0	0	17
PRINCIPAR	7	6	5	0	0	2	1	21
RANLT	7	6	0	0	0	2	1	16
SEXTANT	7	6	5	4	0	0	0	22
TINGYIN	0	0	5	0	0	0	0	5
TOSCA	7	6	0	4	3	2	0	22

Table 3.4: Summary Comparative Evaluation of IPSM Grammatical Annotation Models, WEIGHTED in terms of EAGLES *hierarchy of importance*. Each cell in the table is given a weighted score if the IPSM parsing scheme includes an EAGLES layer (at least partially). *score* is a weighted overall measure of how many layers a parser covers.

outputs to a common notation is a useful exercise, in that it highlights the differences between IPSM parsing schemes. These differences can be categorised according to the EAGLES hierarchy of layers of importance. Table 3.3 in turn highlights the fact that no IPSM parser produces a 'complete' syntactic analysis, and that different parsers output different (overlapping) subsets of the complete picture.

One conclusion is to cast doubt on the value of parser evaluations based purely on success rates, speeds, etc without reference to the complexity of the underlying parsing scheme. At the very least, whatever score each IPSM parser achieves should be modified by a 'parsing scheme coverage' factor: Table 3.3 suggests that, for example, the PRINCIPAR and TOSCA teams should be given due allowance for the richer annotations they attempt to produce. A crude yet topical [3] formula for weighting scores for success rate could be:

*overall-score = success-rate * (parsing-scheme-score - 1)*

However, I assume this formula would not please everyone, particularly the TINGYIN team! This weighting formula can be made even more controvertial by taking the description *heirarchy of importance* at face value, and re-assigning each *yes* cell in Table 3.3 a numerical value on

[3] At the time of writing, UK university researchers are all busy preparing for the HEFCs' Research Assessment Exercise: all UK university departments are to have their research graded on a scale from 5 down to 1. RAE will determine future HEFCs funding for research; a possible formula is: Funding-per-researcher = N*(Grade-1), where N is a (quasi-)constant.

a sliding scale from 7 (a) down to 1 (g), as in Table 3.4. The TOSCA, SEXTANT and PRINCIPAR parsing schemes appear to be "best" as they cover more of the "important" layers of syntactic annotation.

A more useful conclusion is that prospective users of parsers should not take the IPSM parser success rates at face value. Rather, to repeat the point made in Section 3.2, it is not sensible to seek a scale of accuracy applicable across all applications. Different applications require different levels of parsing. Prospective users seeking a parser should first decide what they want from the parser. If they can frame their requirements in terms of the layers of annotation in Table 3.1, then they can eliminate parsers which cannot meet their requirements from Table 3.3. For example, the TOSCA parser was designed for use by researchers in Applied Linguistics and English Language Teaching, who require a complex parse with labelling similar to grammar conventions used in ELT textbooks. In practice, of the IPSM participants only the TOSCA parser produces output suitable for this application, so its users will probably continue to use it regardless of its comparative 'score' in terms of accuracy and speed.

To end on a positive note, this comparative evaluation of grammatical annotation schemes would not have been possible without the IPSM exercise, which generated output from a range of parsers for a common test corpus of sentences. It is high time for more linguists to take up this practical, empirical approach to comparing parsing schemes!

3.10 References

AMALGAM. (1996). WWW home page for AMALGAM. *http://agora.leeds.ac.uk/ccalas/amalgam.html*

Atwell, E. S. (1983). *Constituent Likelihood Grammar ICAME Journal* **7** (34-67). Bergen, Norway: Norwegian Computing Centre for the Humanities.

Atwell, E. S. (1988). Transforming a Parsed Corpus into a Corpus Parser. In M. Kyto, O. Ihalainen & M. Risanen (Eds.) *Corpus Linguistics, hard and soft: Proceedings of the ICAME 8th International Conference* (61-70). Amsterdam, The Netherlands: Rodopi.

Atwell, E. S., Hughes, J. S., & Souter, D. C. (1994). AMALGAM: Automatic Mapping Among Lexico-Grammatical Annotation Models. In J. Klavans (Ed.) *Proceedings of ACL workshop on The Balancing Act: Combining Symbolic and Statistical Approaches to Language* (21-28). New Jersey, USA: Association for Computational Linguistics.

Atwell, E. S., Hughes, J. S., & Souter, D. C. (1994). A Unified MultiCorpus for Training Syntactic Constraint Models. In L. Evett & T. Rose (Eds) *Proceedings of AISB workshop on Computational Lin-*

guistics for Speech and Handwriting Recognition. Leeds, UK: Leeds University, School of Computer Studies.

Atwell, E. S. (1996). Machine Learning from Corpus Resources for Speech And Handwriting Recognition. In J. Thomas & M. Short (Eds.) *Using Corpora for Language Research: Studies in the Honour of Geoffrey Leech* (151-166). Harlow, UK: Longman.

Black, E., Garside, R. G., & Leech, G. N. (Eds.) (1993). *Statistically-driven Computer Grammars of English: the IBM / Lancaster Approach.* Amsterdam, The Netherlands: Rodopi.

comp.speech. (1996). WWW home page for comp.speech Frequently Asked Questions. *http://svr-www.eng.cam.ac.uk/comp.speech/*

EAGLES. (1996). WWW home page for EAGLES. *http://www.ilc.pi.cnr.it/EAGLES/home.html*

HLTsurvey. (1995). WWW home page for the NSF/EC Survey of the State of the Art in Human Language Technology. *http://www.cse.ogi.edu/CSLU/HLTsurvey/*

Hughes, J. S., & Atwell, E. S. (1994). The Automated Evaluation of Inferred Word Classifications. In A. Cohn (Ed.) *Proceedings of European Conference on Artificial Intelligence (ECAI)* (535-539). Chichester, UK: John Wiley.

Hughes, J. S., Souter, D. C., & Atwell, E. S. (1995). Automatic Extraction of Tagset Mappings from Parallel-Annotated Corpora. In E. Tzoukerman & S. Armstrong (Eds.) *From text to tags: issues in multilingual language analysis*, Proceedings of Dublin ACL-SIGDAT workshop. New Jersey, USA: Association for Computational Linguistics.

ICAME. (1996). WWW home page for ICAME. *http://www.hd.uib.no/icame.html*

Johansson, S., Atwell, E. S.,Garside, R. G., & Leech, G. N. (1986). *The Tagged LOB Corpus.* Bergen, Norway: Norwegian Computing Centre for the Humanities.

Leech, G. N., Barnett, R., & Kahrel, P. (1995). *EAGLES Final Report and Guideleines for the Syntactic Annotation of Corpora.* EAGLES Document EAG-TCWG-SASG/1.5 (see EAGLES WWW page) Pisa, Italy: Istituto di Linguistica Computazionale.

O'Donoghue, T. (1993). *Reversing the process of generation in systemic grammar.* PhD Thesis. Leeds, UK: Leeds University, School of Computer Studies.

Sampson, G. (1995). *English for the Computer: the SUSANNE Corpus and Analytic Scheme.* Oxford, UK: Clarendon Press.

Souter, D. C., & Atwell, E. S. (1994). Using Parsed Corpora: A review of current practice. In N. Oostdijk & P. de Haan (Eds.) *Corpus-based Research Into Language* (143-158). Amsterdam, The Netherlands: Rodopi.