Proceedings of the

Workshop on Linguistically Interpreted Corpora

LINC-2000

Edited by

Anne Abeille
Thorsten Brants
Hans Uszkoreit

Held at the Centre Universitaire, Luxembourg, August 6, 2000

Preface

Linguistically interpreted corpora cover a wide variety of topics in Computational Linguistics. They are used for developing processing methods and for linguistic investigations, at the word level and the structural level, for speech and language applications, for syntactic and semantic information, for translation and information retrieval, and for many other topics. More and more types of linguistic information are annotated in corpora.

What was started mainly for English is now performed for many other languages. This allows the transfer of methods previously used for English to other languages as well as the development of new techniques covering needs of other languages. Having corpora in many languages allows to develop and identify techniques applicable to a wide variety of languages. And it allows to adapt to a particular language if necessary.

This workshops aims at bringing together work on linguistic annotation schemes, annotation tools, algorithms aiming at automation of annotation, procedures for efficiently combining human and automatic processes, algorithms aiming at detecting errors and inconsistencies in corpora, search in copora, and the use of corpora for linguistic investigations.

We hope that you will enjoy your time in Luxembourg and find this workshop enjoyable and useful for your work.

Anne Abeille, Department of Linguistics, University Paris 7
Thorsten Brants, Computational Linguistics, Saarland University
Hans Uszkoreit, Computational Linguistics, Saarland University
and German Research Center for AI, Saarbrücken

Workshop on Linguistically Interpreted Corpora

LINC-2000

Luxembourg, August 6, 2000

Organizers:

Anne Abeille (Paris)
Thorsten Brants (Saarbrücken)
Hans Uszkoreit (Saarbrücken)

Programme Committee:

John Carroll (Brighton)
Lionel Clement (Paris)
Tomaz Erjavec (Ljubljana)
Frank Keller (Edinburgh)
Laurent Romary (Nancy)
Geoffrey Sampson (Brighton)
Jean Veronis (Aix-en-Provence)
Atro Voutilainen (Helsinki)
Jakub Zavrel (Antwerp)

Further Information:

Anne Abeille
UFR de Linguistique
Université Paris 7
2, place Jussieu
F-75251 Paris, France
`abeille@linguist.jussieu.fr`

Thorsten Brants, Hans Uszkoreit
FR 8.7 Computational Linguistics
Saarland University
P.O.Box 171150
D-66041 Saarbrücken, Germany
`{brants,uszkoreit}@coli.uni-sb.de`

`http://www.coli.uni-sb.de/linc2000`

Table of Contents

Author Index

Comparing linguistic interpretation schemes for English corpora

Eric ATWELL[1], George DEMETRIOU[2], John HUGHES[3],
Amanda SCHIFFRIN[1], Clive SOUTER[1], Sean WILCOCK[1]

1: School of Computer Studies, University of Leeds, LEEDS LS2 9JT, England;
2: Department of Computer Science, University of Sheffield, SHEFFIELD S1 4DP, England;
3:BT Laboratories, Adastral Park, Martlesham Heath, England
EMAIL: amalgam-tagger@scs.leeds.ac.uk
WWW: http://www.scs.leeds.ac.uk/amalgam/

Abstract

Project AMALGAM explored a range of Part-of-Speech tagsets and phrase structure parsing schemes used in modern English corpus-based research. The PoS-tagging schemes and parsing schemes include some which have been used for hand annotation of corpora or manual post-editing of automatic taggers or parsers; and others which are unedited output of a parsing program. Project deliverables include:

a detailed description of each PoS-tagging scheme, and multi-tagged corpus;
a “Corpus-neutral” tokenization scheme;
a family of PoS-taggers, for 8 PoS-tagsets;
a method for “PoS-tagset conversion”,
a sample of texts parsed according to a range of parsing schemes: a MultiTreebank;
an Internet service allowing researchers worldwide free access to the above resources, including a simple email-based method for PoS-tagging any English text with any or all PoS-tagset(s).

We conclude that the range of tagging and parsing schemes in use is too varied to allow agreement on a standard; and that parser-evaluation based on ‘bracket-matching’ is unfair to more sophisticated parsers.

1. Introduction

The International Computer Archive of Modern and medieval English, ICAME, is an international research network focussing on English Corpus Linguistics, including the collation and linguistic annotation of English language corpora, and applications of these linguistically interpreted corpora. ICAME publishes an annual ICAME Journal (now in its 24th volume) and holds an annual ICAME conference (ICAME’2000, the 19th ICAME conference, was held in Sydney, Australia). Many English Corpus Linguistics projects reported in ICAME Journal and elsewhere involve grammatical analysis or tagging of English texts (eg Leech et al 1983, Atwell 1983, Booth 1985, Owen 1987, Souter 1989a, Benello et al 1989, O’Donoghue 1991, Belmore 1991, Kyto and Voutilainen 1995, Aarts 1996, Qiao and Huang 1998). Each new project reviewed existing tagging schemes, and chose which to adopt and/or adapt.

The project AMALGAM (Automatic Mapping Among Lexico-Grammatical Annotation Models) has explored a range of Part-of-Speech tagsets and parsing schemes used in ICAME corpus-based research. The PoS-tagging schemes include: Brown (Greene and Rubin 1981), LOB (Atwell 1982, Johansson et al 1986), parts (man 1986), SEC (Taylor and Knowles 1988), POW (Souter 1989b), UPenn (Santorini 1990), LLC (Eeg-Olofsson 1991), ICE (Greenbaum 1993), and BNC (Garside 1996). The parsing schemes include some which have been used for hand annotation of corpora or manual post-editing of automatic parsers; and others which are unedited output of a parsing program.

2. Defining the PoS-tagging schemes

ICAME researchers have used a range of different PoS-tag annotation schemes or models. Table 1 shows how an example sentence from the IPSM Corpus (Sutcliffe et al 1996), ‘Select the text you want to protect’, is tagged according to several alternative tagging schemes and vertically aligned.

Table 1 . An example sentence tagged according to eight rival PoS-tagging schemes

	Brown	ICE	LLC	LOB	PARTS	POW	SEC	UPenn
select	VB	V(montr,imp)	VA+0	VB	adj	M	VB	VB
the	AT	ART(def)	TA	ATI	art	DD	ATI	DT
text	NN	N(com,sing)	NC	NN	noun	H	NN	NN
you	PPSS	PRON(pers)	RC	PP2	pron	HP	PP2	PRP
want	VB	V(montr,pres)	VA+0	VB	verb	M	VB	VBP
to	TO	PRTCL(to)	PD	TO	verb	I	TO	TO
protect	VB	V(montr,infin)	VA+0	VB	verb	M	VB	VB
.	.	PUNC(per)	.	.	.	.	.	.

As Corpus Linguists, we preferred to see the tagged corpus as definitive of the meanings and uses of tags in a tagset. We have compiled a detailed description of each PoS-tagging scheme, at a comparable level of detail for each Corpus annotation scheme: a list of PoS-tags with descriptions and example uses from the source Corpus.

We have also compiled a multi-tagged corpus, a set of sample texts PoS-tagged in parallel with each PoS-tagset, and proofread by experts. We selected material from three quite different genres of English (see Table2): informal speech of London teenagers, from COLT, the Corpus of London Teenager English (Andersen and Stenstrom 1996); prepared speech for radio broadcasts, from SEC, the Spoken English Corpus (Taylor and Knowles 1988); and written text in software manuals, from IPSM, the Industrial Parsing of Software Manuals corpus (Sutcliffe et al 1996).

3. A neutral tokenization scheme

An analysis of the different lexical tokenization rules used in the source Corpora has led us to a "Corpus-neutral" tokenization scheme, and consequent adjustments to the PoS-tagsets in our study to accept modified tokenization. The performance of the tagger could be improved by incorporating bespoke tokenisers for each scheme, but we have compromised by using only one for all schemes, to simplify comparisons. This results in errors of the kind exemplified in Table 3, using examples from the POW scheme.

Table 2. Text sources for the multi-tagged corpus.

	Sentences	**Words**	**Average Sentence Length**
London teenager speech (COLT)	60	407	6.8
Radio broadcasts (SEC)	60	2016	33.6
Software manuals (IPSM)	60	1016	16.9
Total:	180	3439	19.1

Table 3. Examples where the standardised tokenizer clashes with a specific tagging scheme (POW)

	Tokeniser/ **Tagger Output**	**Correct analysis** **in POW corpus**
Negatives	are/OM n't/OXN	aren't/OMN
Enclitics	where's/H	where/AXWH 's/OM
Possessives	God's/HN	God/HN 's/G
Expressions	for/P example/H	for-example/A
	have/M to/I	have-to/X

(similarly for set-up, as-well-as, so-that, next-to, Edit/Copy, Drag & Drop, Options... etc.

4. The multi-tagger: a family of PoS-taggers

We trained a publicly-available machine learning system, the Brill tagger (Brill, 1993), to re-tag according to all of the schemes we are working with. As the Brill tagger was the sole automatic annotator for the project we achieved greater consistency. The Brill system is first given a tagged corpus as a training set, from which it extracts a lexicon and two sets of non-stochastic rules: *contextual*, indicating which tag should be chosen in the context of other tags or words, and *lexical*, used to guess the tag for words which are not found in the lexicon. Table 4 shows the model size gleaned from each training set, and accuracy of the re-trained Brill tager on 10,000 words from the source Corpus. The most common errors (as a percentage of all errors for that scheme), are listed in Table 5.

A more realistic evaluation of tagger accuracy across a range of text types was derived in building the multi-tagged corpus, after the outputs of the multi-tagger were proof-read and post-edited by experts in each scheme. Table 6 shows the accuracy of each tagger for the multi-tagged corpus. All the tagging schemes performed significantly worse on this test material than they did on their training material, which indicates how non-generic they are.

Table 4. Model size and accuracy of the re-trained Brill multi-tagger

Tagger	Lexicon	Context Rules	Lexical Rules	Accuracy %
Brown	53113	215	141	97.43
ICE	8305	339	128	90.59
LLC	4772	253	139	93.99
LOB	50382	220	94	95.55
Unix Parts	2842	36	93	95.8
POW	3828	170	109	93.44
SEC	8226	206	141	96.16
Upenn	93701	284	148	97.2

Table 5. The most common PoS-tagging errors.

Brown	VBN/VBD 14.6%	JJ/NN 4.9%	NN/VB 4.2%
ICE	V(cop,pres,encl)/V(intr,pres,encl) 4.1%	ADJ/N(prop,sing) 3.1%	PUNC(oquo)/PUNC(cquo) 2.6%
LLC	PA/AC 4.1%	PA/AP 2.7%	RD/CD 2.7%
LOB	IN/CS 5.8%	TO/IN 4.1%	VBN/VBD 4%
POW	AX/P 4.3%	OX/OM 2.9%	P/AX 2.5%
SEC	TO/IN 6.3%	JJ/RB 5.6%	JJ/VB 4.8%

Table 6. Accuracy found after manual proof-reading of multi-tagged corpus

TAGSET	TOTAL	IPSM60	COLT60	SEC60
Brown	94.3	94.3	87.7	95.6
Upenn	93.1	91.6	88.7	94.6
ICE	89.6	87.0	85.3	91.8
Parts (Unix)	86.7	89.9	82.3	86.0
LLC	86.6	86.9	84.3	87.0
POW	86.4	87.6	87.7	85.4

5. Mapping between tagging schemes

To re-tag the old parts of speech of a corpus with a new scheme of another, we apply our tagger to just the words of the corpus. This might appear to be ‘cheating’; but earlier experiments with devising a set of mapping rules from one tagset to another (Hughes and Atwell 1994, Atwell et al 1994, Hughes et al 1995) concluded that one-to-many and many-to-many mappings predominated over simple one-to-one (and many-to-one) mappings, resulting in more errors than the apparently naïve approach of ignoring the source tags.

6. Comparing tagging schemes

The descriptions of each tagset and multitagged corpus on our website enable corpus-based comparisons between the tagsets. However, quantitative measures are not straightforward. As a simple metric, consider the number of tags in the tagset: this is generally not as simple as it first seems. Most tagsets use tags which are actually a combination of features; this is clearest in ICE (eg **N(com,sing)** for singular common noun), but is also implicit in other tagsets (eg LOB **NN** is also singular common noun, in contrast with **NNS** plural common noun, and **NP** singular proper noun). Our website lists all the tags occurring in the multitagged corpus, but this does not include rare but possible feature-combinations which happen not to occur in the corpus (eg ICE has a tag for plural numbers (as in *three fifths)* which is not used in our corpus). Also, Brown and Upenn tagsets have some tags which are two ‘basic’ tags combined. In Brown, these tags are for enclitic or fused wordforms (eg *I’d* **PPSS+HVD**, *whaddya* **WDT+DO+PPS**); in UPenn, these tags are for words whose analysis is ambiguous or indeterminate (eg *entertaining* **JJ|VBG** = adjective|verb-*ing*-form).

A general observation is that tagsets developed later in time were designed to be ‘improvements’ on earlier tagsets; for example, LOB and UPenn tagsets designers took Brown as a starting-point. So an informal ranking based on age (as given by definitive references) is: Brown (Greene and Rubin 1981), parts (man 1986), LOB (Atwell 1982, Johansson et al 1986), SEC (Taylor and Knowles 1988), POW (Souter 1989b), UPenn (Santorini 1990), LLC (Eeg-Olofsson 1991), ICE (Greenbaum 1993). The ICE tagset is the only one to incorporate explicit features or subcategories, making it more readily digestible by non-expert users : informal feedback from users of our multi-tagger suggests that linguists (and others) find it easier to use tags like **N(com,sing)** than **NN**, since the division into major category and features in brackets is more intuitive. Another class of users of tagged texts are Machine Learning researchers, who want tagged text to train a learning algorithm, but want a small tagset to reduce the problem space; another advantage of the ICE tagset is that it is easy to reduce the tagset to major categories only by ignoring the bracketed features.

7. A MultiTreebank

The differences between English corpus annotation schemes are much greater between parsing schemes for full syntactic structure annotation than they are at word class level. The following are parses of the sentence ‘*Select the text you want to protect.’* according to the parsing schemes of several English parsed corpora or treebanks:

==> ENGCG-BankOfEnglish <==

```
"select" <*>    V IMP VFIN  @+FMAINV
"the"  DET CENTRAL ART SG/PL  @DN>
"text" N NOM SG @OBJ
"you"  PRON PERS NOM SG2/PL2 @SUBJ
"want"    V PRES -SG3 VFIN  @+FMAINV
"to" INFMARK>  @INFMARK>
"protect"  V INF @-FMAINV
"<$.>"
```

==> L-ATR <==

```
[start [sprpd1 [sprime1 [sc3 [v2
Select_VVIOPTATIVE [nbarq14 [nbar4 [d1
the_AT d1] [n1a text_NN1MESSAGE n1a] nbar4]
[sd1 [nbar6 you_PPY nbar6] [vbar1 [v33
want_VVOMENTAL-ACT [vibar1 to_TO [v1
protect_VVIHELP v1] vibar1] v33] vbar1] sd1]
nbarq14] v2] sc3] sprime1] ._. sprpd1] start]
```

==> ***LOB-TREEBANK*** <==

```
[S[V[VB[ select ]VB]V][Ns[ATI[ the ]ATI][NN[
text ]NN][Fr[N[PP2[ you ]PP2]N][V[VB[ want
]VB]V][Ti[Vi[TO[ to ]TO][VB[ protect
]VB]Vi]Ti]Fr]Ns][.[ . ].]S]
```

==> ***POW*** <==

```
Z CL 1 M SELECT 1 C NGP 2 DD THE 2 H
TEXT 2 Q CL 3 S NGP HP YOU 3 M WANT 3 C
CL 4 I TO 4 M PROTECT 1 ? .
```

==> ***SEC*** <==

```
[V Select_VV0 [N the_AT text_NN1 [Fr[N
you_PPY N][V want_VV0 [Ti to_TO
protect_VV0 Ti]V]Fr]N]V] ._.
```

==> ***SUSANNE*** <==

```
VV0t Select select [O[S*[V.V]
AT the the [Ns:o101.
NN1n text text .
PPY you you[Fr[Ny:s103.Ny:s103]
VV0v want want [V.V]
YG - - [Ti:o[s103.s103]
TO to to [Vi.
VV0t protect protect .Vi]
YG — -
[o101.o101]Ti:o]Fr]Ns:o101]S*]
YF +. - O]
```

There ae two main approaches to format : one word per line, with parsing annotations (ENGCG, SUSANNE), aimed at human proofreaders, to make it easier to scan parses and correct errors; and tree-structure captured via lisp-like bracketting (L-ATR, LOB-TREEBANK, SEC, POW), assuming the textfile is processed by a tree-viewing program for human end-user consumption. The POW format uses a numerical code capable of capturing crossing branches, but in principle encodes the phrase structure.

There is even greater diversity in the parsing schemes (and formats) used in alternative NLP parsing *programs*. The example sentence was actually selected from a test-set used at the Industrial Parsing of Software Manuals workshop (Sutcliffe et al 1996); it is one of the shortest test sentences, which one might presume to be one of the most grammatically straightforward and uncontroversial. The following are outputs of several rival NLP parsing programs, given the example sentence to parse:

==> ***alice*** <==

```
Fragment No. 1
>From 0 To 5
(SENT (SENT-MOD (UNK-CAT "Select") (NP
(DET "the") (NOUN "text")))
(SENT (VP-ACT (NP "you") (V-TR "want")) (NP
NULL-PHON)))
Fragment No. 2
>From 5 To 7
(SENT-MOD (UNK-CAT "to") (NP "protect"))
```

==> ***despar*** <==

```
VB select 1 --> 8 -
DT the 2 --> 3 [
NN text 3 --> 1 + OBJ
PP you 4 --> 5 " SUB
VBP want 5 --> 3 ]
TO to 6 --> 7 -
VB protect 7 --> 5 -
. . 8 --> 0 -
```

==> ***principar_constituency*** <==

```
(S
(VP (Vbar (V (V_NP
(V_NP Select)
(NP
(Det the)
(Nbar
(N text)
(CP
Op[1]
(Cbar (IP
(NP (Nbar (N you)))
(Ibar (VP (Vbar (V (V_CP
(V_CP want)
(CP (Cbar (IP
PRO
(Ibar
(Aux to)
(VP (Vbar (V (V_NP
(V_NP protect)
t[1]))))))))))))))))))))))
.)
```

==> ***principar_dependency*** <==

```
(
(Select ~ V_NP *)
(the ~ Det < text spec)
(text ~ N > Select comp1)
(you ~ N < want subj)
(want ~ V_CP > text rel)
(to ~ I > want comp1)
(protect ~ V_NP > to pred)
(. )
)
```

==> *ranlp* <==

```
(VP/NP select
(N2+/DET1a the
(N2-
(N1/INFMOD
(N1/RELMOD1 (N1/N text)
(S/THATLESSREL (S1a (N2+/PRO you) (VP/NP
want (TRACE1 E)))))
(VP/TO to (VP/NP protect (TRACE1 E)))))))
```

==> *sextant* <==

```
134 ------------------
Select the text you want to protect .
134 VP 101 Select select INF 0 0
134 NP 2 the the DET 1 1 2 (text) DET
134 NP* 2 text text NOUN 2 1 0 (select) DOBJ
134 NP* 3 you you PRON 3 0
134 VP 102 want want INF 4 0
134 VP 102 to to TO 5 0
134 VP 102 protect protect INF 6 1 3 (you) SUBJ
134 -- 0 . . . 7 0
```

This sentence is part of our multi-parsed corpus or MultiTreebank (Atwell 1996). The parsing schemes exemplified in our MultiTreebank include some which have been used for hand annotation of corpora or manual post-editing of automatic parsers: EPOW (O'Donoghue 1991), ICE (Greenbaum 1992), POW (Souter 1989a,b), SEC (Taylor and Knowles 1988), and UPenn (Marcus et al 1993). Linguist experts in each of these corpus annotation schemes kindly provided us with their parsings of the 60 IPSM sentences. Others are unedited output of parsing programs: Alice (Black and Neal 1996), Carroll/Briscoe Shallow Parser (Briscoe and Carroll 1993), DESPAR (Ting and Shiuan 1996), ENGCG (Karlsson et al 1995, Voutilainen and Jarvinen 1996), Grammatik (WordPerfect 1998), Link (Sleator and Temperley 1991, Sutcliffe and McElligott 1996), PRINCIPAR (Lin 1994, 1996), RANLP (Osborne 1996), SEXTANT (Grefenstette 1996), and TOSCA (Aarts et al 1996, Oostdijk 1996). Language Engineering researchers working with these systems kindly provided us with their parsings of the 60 IPSM sentences.

The MultiTreebank illustrates the diversity of parsing schemes available for modern English language corpus annotation. The (EAGLES 1996) guidelines recognise layers of syntactic annotation, which form a hierarchy of importance. None of the parsing schemes included here contains all the layers (*a-h*, in Table 7 below). Different parsers annotate with different subsets of the hierarchy.

7. Website and e-mail tagging service

The multi-tagged corpus, multiTreebank, tagging scheme definitions and other documentation are available on our website. Email your English text to *amalgam-tagger@scs.leeds.ac.uk*, and it will be automatically processed by the multi-tagger, and then the output is mailed back to you. Users can select any or all of the eight schemes (Brown, ICE, LLC,LOB, Parts, POW, SEC, UPenn). The tagged text is returned one email reply message per scheme. A verbose mode can also be selected, which gives the long name for each tag as well as its short form in the output file.

Table 7. Evaluation of MultiTreebank parse schemes in terms of EAGLES layers of syntactic annotation :
(a) Bracketing of segments
(b) Labelling of segments
(c) Showing dependency relations
(d) Indicating functional labels
(e) Marking sub-classification of syntactic segments
(f) Deep or 'logical' information
(g) Information about the rank of a syntactic unit
(h) Special syntactic characteristics of spoken language

Parse Scheme	EAGLES layer								
	a	b	c	d	e	f	g	h	Score
ALICE	yes	yes	no	no	no	no	no	no	2
CARROLL	yes	yes	no	no	no	no	no	no	2
DESPAR	no	no	yes	no	no	no	no	no	1
ENGCG	no	no	yes	yes	yes	no	no	no	3
EPOW	yes	yes	no	yes	no	no	no	yes	4
GRAMMATIK	yes	yes	no	yes	no	no	no	no	3
ICE	yes	yes	no	yes	yes	no	no	yes	5
LINK	no	no	yes	yes	no	no	no	no	2
POW	yes	yes	no	yes	no	yes	no	yes	5
PRINCIPAR	yes	yes	yes	no	no	yes	yes	no	5
RANLT	yes	yes	no	no	no	yes	yes	no	4
SEC	yes	yes	no	no	yes	no	no	yes	4
SEXTANT	yes	yes	yes	yes	no	no	no	no	4
TOSCA	yes	yes	no	yes	yes	yes	no	yes	6
UPENN	yes	yes	no	no	no	No	no	no	2

The service has been running since December 1996, and usage is logged on our website; up to December 1999, it processed 19,839 email messages containing over 628 megabytes of text. The most popular schemes are LOB, UPenn, Brown, ICE, and SEC (in that order), with relatively little demand for parts, LLC, and POW; this reflects the popularity of the source corpora in the Corpus Linguistics community. Apart from obvious uses in linguistic analysis, English language teaching and learning, and teaching Natural Language Processing and Artificial Intelligence university students, some unforeseen applications have been found, e.g. in using the tags to aid data compression of English text (Teahan 1998); and as a guide in the search for extra-terrestrial intelligence (Elliott and Atwell 2000, Elliott et al 2000).

8. Conclusions

NLP researchers have not agreed a standard lexico-grammatical annotation model for English, so the AMALGAM project has investigated a range of alternative schemes. We have trained a 'machine learning' tagger with several lexico-grammatical annotation models, to enable it to annotate according to several rival modern English languge corpus Part-of-Speech tagging schemes. Our main achievements are:

Software: *PoS-taggers* trained to annotate text according to several rival lexico-grammatical annotation models, accessible over the Internet via email.

Data-sets: a *multi-tagged corpus* and *multi-treebank*, a corpus of English text where each sentence is annotated according to several rival lexico-grammatical annotation models. We have also collected together definitions of eight major English corpus word-tagging schemes. All are available over the Internet via WWW.

We conclude that there is still work to be done on agreeing a truly generic PoS-tagging scheme; and that it is not possible, to map between all parsing schemes. Unlike the tagging schemes, it does not make sense to make an application-independent comparative evaluation. No single standard can be applied to all parsing projects. Even the presumed lowest common denominator, bracketing, is rejected by some corpus linguists and dependency grammarians. The guiding factor in what is included in a parsing scheme appears to be the author's theoretical persuasion or the application they have in mind.

Acknowledgements

We are grateful to the UK Engineering and Physical Sciences Research Council (EPSRC) for funding this research project. We are also indebted to the numerous researchers worldwide who helped us by providing advice, data, and documentation, and by proofreading the multi-tagged Corpus and MultiTreebank.

This COLING Workshop paper is an an abridged version of a full paper published in ICAME Journal, (Atwell et al 2000); we are grateful for the Journal's permission to present our findings to this complementary Workshop audience. To get the full ICAME Journal paper, see http://www.hd.uib.no/icame/journal.html

References

Aarts, Jan. 1996. A tribute to W. Nelson Francis and Henry Kucera: grammatical annotation. *ICAME Journal* 20:104-107.

Aarts, Jan, Hans van Halteren and Nelleke Oostdijk. 1996. The TOSCA analysis system. In C. Koster and E Oltmans (eds). *Proceedings of the first AGFL workshop.* 181-191. Technical Report CSI-R9604, Computing Science Institute, University of Nijmegen.

Andersen, Gisle, and Anna-Brita Stenstrom. 1996. COLT: a progress report. *ICAME Journal* 20:133-136.

Atwell, Eric. 1982. *LOB Corpus tagging project: post-edit handbook.* Department of Linguistics and Modern English Language, University of Lancaster.

Atwell, Eric. 1983. Constituent Likelihood grammar. *ICAME Journal* 7:34-66.

Atwell, Eric. 1996. Comparative evaluation of grammatical annotation models. In (Sutcliffe et al 1997), 25-46.

Atwell, Eric, John Hughes and Clive Souter. 1994. AMALGAM: Automatic Mapping Among Lexico-Grammatical Annotation Models. In Judith Klavans and Philip Resnik (eds.), *The balancing act - combining symbolic and statistical approaches to language. Proceedings of the workshop in conjunction with the 32nd annual meeting of the Association for Computational Linguistics.* New Mexico State University, Las Cruces, New Mexico, USA.

Atwell, Eric, George Demetriou, John Hughes, Amanda Schiffrin, Clive Souter, and Sean Wilcock. 2000. A comparative evaluation of modern English corpus grammatical annotation schemes. *ICAME Journal* 24, to appear.

Belmore, Nancy. 1991. Tagging Brown with the LOB tagging suite. *ICAME Journal* 15:63-86.

Benello, J., A. Mackie and J. Anderson. 1989. Syntactic category disambiguation with neural networks. *Computer Speech and Language* 3:203-217.

Black, William, and Philip Neal. 1996. Using ALICE to analyse a software manual corpus. In (Sutcliffe et al 1996), 47-56.

Booth, Barbara. 1985. Revising CLAWS. *ICAME Journal* 9:29-35.

Brill, Eric. 1993. *A Corpus-based approach to language learning.* PhD thesis, Department of Computer and Information Science, University of Pennsylvania.

Briscoe, Edward and John Carroll. 1993. Generalised probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics* 19:25-60.

EAGLES (1996), WWW site for European Advisory Group on Language Engineering Standards, http://www.ilc.pi.cnr.it/EAGLES96/home.html Specifically: Leech, Geoffrey, Ruthanna Barnett and Peter Kahrel, *EAGLES Final Report and guidelines for the syntactic annotation of corpora*, EAGLES Report EAG-TCWG-SASG/1.5.

Eeg-Olofsson, Mats. 1991. *Word-class tagging: Some computational tools*. PhD thesis. Department of Linguistics and Phonetics, University of Lund, Sweden.

Elliott, John, and Eric Atwell. 2000. Is there anybody out there?: the detection of intelligent and generic language-like features. In *Journal of the British Interplanetary Society*, 53:1/2, 13-22.

Elliott, John, Eric Atwell, and Bill Whyte. 2000. Language identification in unknown signals. Proc COLING'2000.

Garside, Roger. 1996. The robust tagging of unrestricted text: the BNC experience. In Jenny Thomas and Mick Short (eds) *Using corpora for language research: studies in the honour of Geoffrey Leech*, 167-180. London: Longman.

Greene, Barbara and Gerald Rubin. 1981. *Automatic grammatical tagging of English*. Providence, R.I.: Department of Linguistics, Brown University.

Greenbaum, Sidney. 1993. The tagset for the International Corpus of English. In Clive Souter and Eric Atwell (eds) *Corpus-based Computational Linguistics*. 11-24. Amsterdam: Rodopi.

Grefenstette, Gregory. 1996. Using the SEXTANT low-level parser to analyse a software manual corpus. In (Sutcliffe et al 1996), 139-158.

Hughes, John and Eric Atwell. 1994. The automated evaluation of inferred word classifications. In Anthony Cohn (ed.), *Proceedings of the European Conference on Artificial Intelligence (ECAI)*. 535-539. Chichester, John Wiley.

Hughes, John, Clive Souter and Eric Atwell. 1995. Automatic extraction of tagset mappings from parallel-annotated corpora. In *From texts to tags: issues in multilingual language analysis. Proceedings of SIGDAT workshop in conjunction with the 7th Conference of the European Chapter of the Association for Computational Linguistics.* University College Dublin, Ireland.

Johansson, Stig, Eric Atwell, Roger Garside and Geoffrey Leech. 1986. *The Tagged LOB corpus: users' manual.* Bergen University, Norway: ICAME, The Norwegian Computing Centre for the Humanities. Available from http://www.hit.uib.no/icame/lobman/lob-cont.html

Karlsson, Fred, Atro Voutilainen, Juha Heikkila, and Arto Anttila. 1995. *Constraint Grammar: a language-independent system for parsing unrestricted text.* Berlin: Mouton de Gruyter.

Kyto, Merja and Atro Voutilainen. 1995. Applying the Constraint Grammar parser of English to the Helsinki corpus. *ICAME Journal* 19:23-48.

Leech, Geoffrey, Roger Garside and Eric Atwell. 1983. The automatic grammatical tagging of the LOB corpus. *ICAME Journal* 7:13-33.

Lin, Dekang. 1994. PRNCIPAR – an efficient, broad-coverage, principle-based parser. *Proceedings of COLING-94, Kyoto*. 482-488.

Lin, Dekang. 1996. Using PRINCIPAR to analyse a software manual corpus. In (Sutcliffe et al 1996), 103-118.

man 1986. *parts*. The on-line Unix manual.

Marcus, Mitch, M Marcinkiewicz, and Barbara Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics* 19:313-330.

O'Donoghue, Tim. 1991. Taking a parsed corpus to the cleaners: the EPOW corpus. *ICAME Journal* 15:55-62

Oostdijk, Nelleke. 1996. Using the TOSCA analysis system to analyse a software manual corpus. In (Sutcliffe et al 1996), 179-206.

Osborne, Miles. 1996. Using the Robust Alvey Natural Language Toolkit to analyse a software manual corpus. In (Sutcliffe et al 1996), 119-138.

Owen, M. 1987. Evaluating automatic grammatical tagging of text. *ICAME Journal* 11:18-26.

Qiao, Hong Liang and Renje Huang. 1998. Design and implementation of AGTS probabilistic tagger. *ICAME Journal* 22: 23-48.

Santorini, Barbara. 1990. *Part-of-speech tagging guidelines for the Penn Treebank project.* Technical report MS-CIS-90-47. University of Pennsylvania: Department of Computer and Information Science.

Sleator, D. and Temperley, D. 1991. *Parsing English with a Link grammar*. Technical Report CMU-CS-91-196. School of Computer Science, Carnegie Mellon University.

Souter, Clive. 1989a. The COMMUNAL project: extracting a grammar from the Polytechnic of Wales corpus. *ICAME Journal* 13:20-27.

Souter, Clive. 1989b *A short handbook to the Polytechnic of Wales Corpus*. Bergen University, Norway: ICAME, The Norwegian Computing Centre for the Humanities. Available from http://kht.hit.uib.no/icame/manuals/pow.html

Sutcliffe, Richard, Heinz-Detlev Koch and Annette McElligott (eds.). 1996. *Industrial parsing of software manuals*. Amsterdam: Rodopi.

Sutcliffe, Richard, and Annette McElligott. 1996. Using the Link parser of Sleator and Temperley to analyse a software manual corpus. In (Sutcliffe et al 1996), 89-102.

Taylor, Lolita and Gerry Knowles. 1988. *Manual of information to accompany the SEC corpus: The machine readable corpus of spoken English.* University of Lancaster: Unit for Computer Research on the English Language. Available from

http://kht.hit.uib.no/icame/manuals/sec/INDEX.HTM

Teahan, Bill. 1998. *Modelling English text.* PhD Thesis, Department of Computer Science, University of Waikato, New Zealand.

Ting, Christopher, and Peh Li Shiuan. 1996. Using a dependency structure parser without any grammar formalism to analyse a software manual corpus. In (Sutcliffe et al 1996), 159-178.

Voutilainen, Atro, and Timo Jarvinen. 1996. Using the English Constraint Grammar Parser to analyse a software manual corpus. In (Sutcliffe et al 1996), 57-88.

Dependency-based Syntactic Annotation of a Chinese Corpus

Tom B.Y. LAI
City University of Hong Kong
Tsinghua University, Beijing
cttomlai@cityu.edu.hk

HUANG Changning
Microsoft Research, China
cnhuang@microsoft.com

Abstract

We discuss a syntactic annotation scheme for Chinese text corpora following a dependency-based framework that admits no intermediate phrasal nodes and allows no crossing of syntactic dependency links. While one particular approach to syntactic analysis is being followed, dependency annotation facilitates the use of annotated corpora by followers of other approaches.

1 Introduction

Major linguistic theories like GB/MP (Chomsky, 1986; Chomsky, 1995), HPSG (Pollard and Sag, 1994) and LFG (Bresnan, 1982) agree to represent syntactic structures in terms of phrase structures, but disagree about what kinds of phrase structures should be assigned to the same linguistic expressions. In a project on syntactic annotation of Chinese corpora, we represent syntactic structures in terms of dependency. [1]

We follow an approach (Lai and Huang, 1998a; Lai and Huang, 1999a) to Dependency Grammar (Tesnière, 1959; Gaifman, 1965; Hays, 1964; Robinson, 1970), that requires syntactic dependency to be single-headed and projective. Unlike Dependency Grammar schools that allow multiple-headed and non-projective dependency structures (Hudson, 1984; Mel'čuk, 1988; Starosta, 1988; Hajičova, 1991), single-headedness and projectivity are maintained in a syntactic skeleton, with reference to which constraints to capture non-projective phenomena in language are anchored. In experimental implementations (Lai and Huang, 1998b; Lai and Huang, 1999b) of this approach, projective syntactic dependency structures are generated subject to the constraints of subcategorization properties of the words concerned as well as other grammatical considerations. This approach is different from many works in dependency-based parsing (Hellwig, 1986; Covington, 1990; Courtin and Genthial, 1998; Bourdon et al., 1998) in that the relationship between the governor and all its dependents are immediate and no intermediate phrasal nodes are necessary. Similar "flat" syntactic structures have recently been suggested in phrase-structure grammars (Bouma et al., 1998; Przepiórkowski, 1999).

In preparation for large-scale annotation, we are carrying out manual syntactic annotation of a small Chinese legal text corpus. The text is first processed using a "segmentation" and "tagging" tool (Lai et al., 1992; Lai et al., 1998). The tokens are then subjected to morphological analysis to confirm and adjust word boundaries. Words, as the units that are operated on in syntactic analysis, form the basis of an SGML-based annotation scheme. The annotation scheme also recognizes larger parsing units like phrases and sentences and smaller units like *characters* and dictionary entries, which may or may not coincide with the words.

Following accepted practices of text corpora annotation, the original character sequences of the raw corpus are preserved as the terminally tagged elements. This enables recovery from possible errors in morphological and syntactic analysis. The representation of syntactic relationships in terms of dependency also facilitates the use of the annotated corpus by followers of other approaches.

2 Projective dependency syntax without intermediate phrasal nodes

2.1 Projective syntactic dependency skeleton

In Dependency Grammar (Tesnière, 1959), words are linked to one another by asymmetrical governor-dependent relationships. Syntactic dependency structures are constrained by Robinson (1970) as follows:

(1) a. One and only one element is independent.

b. All others depend directly on some element.

c. No element depends directly on more than one other.

[1] The Academica Sinica (Taipei) Chinese treebank and the LDC Chinese Treebank Project should be noted.

d. If A depends directly on B and some element C intervenes between them (in linear order of string), then C depends directly on A or on B or some other intervening element.

Robinson requires that a word should not depend on more than one word. She also requires that syntactic dependency structures be *projective* in the sense that dependency links should not cross one another.

For example, the projectivity criterion will be violated if the word *ta* ('he') in the Chinese sentence (2) is considered to depend on the matrix verb *xiang* ('wanted') and the embedded verb *xiao* ('laugh') at the same time.

(2) Ta xiang xiao.

he want laugh

'He wants/wanted to laugh.'

In Figure 1, the dependency link between *ta* and *xiao* crosses the branch linking the matrix verb *xiang* to the root node. This situation is dealt with by suggesting a projective skeleton structure as in Figure 2. The link between *ta* and *xiao* is severed. The fact that *ta* is the subject of *xiao* is accounted for by a specification in the lexical entry of the word *xiang*: the subject of *xiang* is the subject of its predicate complement *xiao*. Other non-projective linguistic phenomena are dealt with similarly by grammatical constraints defined in terms of the nodes and arcs of the projective syntactic dependency skeleton.

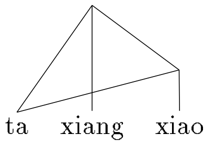


Figure 1: Non-projective syntactic structure

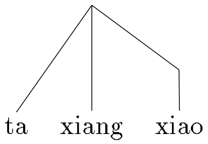


Figure 2: Projective syntactic structure

2.2 Dependency rules

Projective dependency structures can be generated using Hays' (1964) dependency rules.

(3) a. X(A, B, C, ..., *, Y, ..., Z)

b. X(*)

c. *(X)

In (3), dependents of the governor X are listed between a pair of brackets, with the asterisk * indicating the position of the governor itself.

Hays' rules have the disadvantage of having word order (of dependents with the same governor) built into the rule mechanism. This disadvantage is removed by making dependency rules binary-branching.

Repeated application of binary-branching dependency rules will over-generate, but subcategorization properties of the governing word and global grammatical constraints of the language will co-operate to function as a filter and account for the correct ensemble of dependent elements in the "domain" of the governor.

In dependency rules, the governor and its placeholder * are not only of the same *type*, as in phrase-structure rewrite rules, but also *token*-identical. The result is that a "phrase" is indistinguishable from its head word, and the ensemble of a head word and its dependent is a "flat" structure without intermediate phrasal nodes.

3 Dependency-based annotation

3.1 A small text corpus

We begin with manually annotating a small corpus, with a plan to scale up with the help of the experience gained. We use a small corpus of Chinese text segmented and tagged using a bigram-based segmentation-tagging tool (Lai et al., 1998). The corpus is two "chapters" of a statute in an East Asian Chinese community (Hong Kong). It contains 4797 tokens produced by the segmentation-tagging tool. Because of its small size, this corpus is stylistically not balanced, which is to be borne in mind.

3.2 SGML-style annotation scheme

The annotation scheme is based on SGML (SGML, 1986). Its design is explained with the help of the following example:

```
<pu pi=1>
<mu mi=i wu=1><du tg=hm><cu>"di4"
<wu wi=1 gv=2 fn=nm ct=mx sm="seventh">
  <du tg=mx ><cu>"qi1"
<wu wi=2 gv=0 ct=ncl sm="chapter">
  <du tg=cnb><cu>"zhang1"
</pu>
<pu pi=2>
<wu wi=1 gv=0 fn=sub ct=nc mh="de"
    sm="contract">
  <du tg=ncd><cu>"he2"<cu>"yue4"
<mu mi=1 wu=1><du tg=ed><cu>"de"
<wu wi=2 gv=0 fn=sub ct=na sm="form">
  <du tg=nad><cu>"xing2"<cu>"shi4"
```

```
</pu>
<pu pi=3>
<wu wi=1 gv=0 ct=mx sm="7.1">
  <du tg=mx><cu>7<du tg="."><cu>.
  <du tg=mx><cu>1
</pu>
<pu pi=4>
<wu wi=1 gv=0 ct=na sm="form">
  <du tg=nad><cu>"xing2"<cu>"shi4"
</pu>
<pu pi=5>
<wu wi=1 gv=0 ct=mx sm="7.1.1">
 <du tg=mx><cu>7<du tg="."><cu>.
 <du tg=mx><cu>1<du tg="."><cu>.
 <du tg=mx><cu>1
</pu>
<pu pi=6>
<wu wi=1 gv=7 fn=sub ct=nc sm="contract">
  <du tg=ncd><cu>"he2"<cu>"yue4"
<wu wi=2 gv=5 fn=mks ct=cnj sm="because">
  <du tg=jom><cu>"yin1"
<wu wi=3 gv=5 fn=sub ct=na sm="form">
  <du tg=nad><cu>"xing2"<cu>"shi4"
<wu wi=4 gv=5 fn=neg ct=adv sm="neg">
  <du tg=bu><cu>"bu4"
<wu wi=5 gv=7 fn=ajt ct=vt sm="conform">
  <du tg=vnm><cu>"he2"
<wu wi=6 gv=5 fn=obj ct=na
    sm="stipulation">
  <du tg=nad><cu>"gui1"<cu>"ge2"
<wu wi=7 gv=0 ct=aux sm="can">
  <du tg=ud><cu>"ke3"<cu>"neng2"
<wu wi=8 gv=7 fn=axo ct=aj sm="invalid">
  <du tg=aod><cu>"wu2"<cu>"xiao4"
<wu wi=9 gv=11 fn=mkc ct=cnj sm="or">
  <du tg=jom><cu>"huo4"
<wu wi=10 gv=11 fn=neg ct=av sm="neg">
  <du tg=bu><cu>"bu4"
<wu wi=11 gv=8 fn=cjt ct=aux sm="can">
  <du tg=um><cu>"neng2"
<wu wi=12 gv=11 fn=axo ct=vt
    sm="carry out">
  <du tg=vnd><cu>"li3"<cu>"xing2"
<wu wi=13 gv=7 fn=pt ct="..">
  <du tg=".."><cu>..
</pu>
```

The terminal text elements, marked by the $< cu >$ tags, are Chinese (*Han*) characters in a two-byte encoding scheme. They are written in the phonetic *pingyin* script in this paper for the benefit of the reader.

The largest text unit for syntactic analysis shown here is not the *sentence*, but the *parsing unit* $< pu >$. There are six such units in the example: an ordinal numerical phrase, a chapter title, two numeral construction in Arabic numberals, a section heading, and a one-sentence subsection text.

The words, as the basic units of subsequent syntactic analysis, play a key role in the annotation scheme. The semantic glosses of the words are given in the *sm* attribute. In general, sub-word morphological units are contained within the scope of a $< wu >$ tag. The $< du >$ tag marks dictionary entries as a sub-word units, which, especially in Chinese, often do not coincide with the words they constitute.

The tags $< wu >$ and $< cu >$ correspond to the $< w >$ and $< c >$ tags for *linguistic segmentation elements* in CES (Ide et al., 1996). The usage of $< mu >$, however, is different from that of $< m >$ in CES. We do not have tags corresponding to $< cl >$ and $< phr >$ in CES. As noted earlier, $< pu >$ can be a word, a phrase or a sentence.

The $< wu >$ elements have an index attribute wi to mark their positions within the $< pu >$ unit. They also have a gv attribute recording the wi indices of their governors. A value of 0 shows that the word is the head element of the $< pu >$. The syntactic category of the word is given by the ct attribute, and its relation to its immediate governor is the fn attribute.

When values are assigned to the gv attribute of $< wu >$, care is taken to have Robinson's "axioms" of well-formedness (1) observed. Projectivity is ensured by checking that the gv value of a word is neither smaller than that of any words preceding it in the same $< pu >$ nor greater than that of any other words following it in the $< pu >$.

In our annotation scheme, morphemes (mu) are marked only when they are not adequately covered by the words and the dictionary entries. When they morphemes are marked, as in $< pu\ pi = 1 >$ and $< pu\ pi = 2 >$, they are not marked as constituents of a wu. This will be explained later in this paper.

4 Basic features of the annotation scheme

4.1 Preservation of raw text elements

Chinese texts are stored as sequences of "characters" without explicit word boundary marks. With very few exceptions, Chinese characters are meaning-bearing syllables. They may function either as one-morpheme words or as morphemes that combine to form words. Unfortunately, Chinese linguists do not always agree about how a given sequence of characters should be "segmented" in words. It is thus important that the raw character sequence of the original text should be preserved for the benefit of people who do not agree with us.

The basic encoding units of European texts are the letters of the alphabet. Letters combine to form words, which are marked off from one another by white spaces (though sometimes "words" will have to be combined to form compound words). It is thus not uncommon for the terminally tagged units of annotated European texts to be (lemmatized) words. This will be fine if the morphological analysis is always correct, but will make recovery from errors like mistaking "bake[PAST]" for "bake[PART]" difficult.

In our annotation scheme, punctuation marks are also preserved and marked as such in our annotation scheme. Besides providing hints for syntactic and pragmatic analysis, they also mark off small chunks of character sequences for the segmentation-tagging program to operate on.

4.2 Flexibility of parsing units

In Chinese test, as well as in texts in other languages, the chunk of text that one has to feed into a "sentence" analyzer are often not a sentence. In the example in the previous section, $< pu\ pi = 1 >$, $< pu\ pi = 2 >$, $< pu\ pi = 3 >$, $< pu\ pi = 4 >$ and $< pu\ pi = 5 >$ are all not sentences. In an English translation of the text, they may be rendered as *Chapter Seven*, *The Form of a Contract*, *7.1*, *Form*, and *7.1.1* respectively, which should also be treated as non-sentence parsing units.

Thus, we allow parse units to be anything suggested by the text itself. shown in the example, they may be words, phrases and, of course, sentences. Parse units form separate domains for the position indices of their constituent words. Head words of parse units are not assigned governors of of their own. Relationships between parse units are considered to belong to the realm of pragmatics.

4.3 Dependency marking

We do not mark phrase structures. There no bracketing as in the PENN Treebank. There are no intermediate phrase nodes as in GB/MP, HPSG and LFG, and the relationship between governor and dependent is always direct. This does not not solve the problem of different approaches producing different syntactic structures for the same linguistic expression, but rather accentuates it in a somewhat positive sense. This will be discussed in greater detail in a later section.

5 Morphological complications

5.1 Deriving words from dictionary units

In Chinese, as in all other languages, words may combine to form larger compounds words. It is often justified to have compound words listed in our dictionary. Sometimes, however, listing a word formed from simpler words in a dictionary can be unrealistic or unreasonable.

The third $< pu >$ in Section 3.2, repeated below, serves to illustrate this.

```
<pu pi=3>
<wu wi=1 gv=0 ct=mx sm="7.1">
  <du tg=mx><cu>7<du tg="."><cu>.
  <du tg=mx><cu>1
</pu>
```

The segmentation-tagging program outputs three "tokens" as candidates for separate words. We adjust the word boundaries and group the three characters together to form only one word ($< wu >$), which is a kind or numerical label.

We do not consider this an error of the segmentation-tagging program. Numeral characters like *7*, "." and *1* are dictionary entries that are, with the helps of marks like the point ".", capable of combining transparently to form an infinite number of words like *7.1*, which is not, and cannot all be, listed in a dictionary. In the example, we mark the three characters as a word, but also retain the information that this word is composed from the three one-character dictionary entries. The grammatical information originally attached to the three constituent "tokens" are retained. They are tagged as dictionary entries ($< du >$).

Obviously, word units like *7.1* are also possible in other languages. Besides, in Chinese and in other languages, it is sometimes difficult to decide whether a number of space-separated "words" should combine to form a compound word. In view of this, dictionary unit tags are a good way to ensure the usefulness of the annotated corpus.

5.2 Derivational and inflectional affixes

Derivational morphology is encountered in first $< pu >$ in the example in Section 3.2:

```
<pu pi=1>
<mu mi=i wu=1><du tg=hm><cu>"di4"
<wu wi=1 gv=2 fn=nm ct=mx sm="seventh">
  <du tg=mx><cu>"qi1"
<wu wi=2 gv=0 ct=ncl sm="chapter">
  <du tg=cnb><cu>"zhang1"
</pu>
```

In *di4qi1*, the prefix *di4* is attached to the cardinal number *qi1* ('seven') to turn it into an ordinal number. Like *-th* and *-ieme* in English and French, *di4* is a bound morpheme in modern Chinese. However, this prefix is listed as a separate entry in all Chinese dictionaries, and the ordinary native speaker has difficulty in seeing it as different from "real" words in the language. Anyway, no graphical hints are available to distinguish between free and bound morphemes in Chinese text.

In respect of the rather general practice of the Chinese computational linguistics community to mark off bound grammatical morphemes like *di*4 as separate "words". affixes are marked as $< mu >$ and placed immediately under $< pu >$ like $< wu >$. However, as we choose to consider affixes as part of the words to which they are attached, they are not given an independent position index, and their *mi* indices are meaningful only within the scope of the words to which they are attached. The syntactic categories and the meanings of the word units are those of the derived words.

The second < *pu* > is an example of inflectional morphology.

```
<pu pi=2>
<wu wi=1 gv=0 fn=sub ct=nc mh="de"
   m="contract">
  <du tg=ncd><cu>"he2"<cu>"yue4"
<mu mi=1 wu=1><du tg=ed><cu>"de"
<wu wi=2 gv=0 fn=sub ct=na sm="form">
  <du tg=nad><cu>"xing2"<cu>"shi4"
</pu>
```

Inflectional affixes are dealt with like derivational affixes. The suffix *de* is a genitive marker in Chinese. It is marked as an < *mu* > and is assigned an *mi* and assigned an *mi* index that is only meaningful to the word *he2yue4* ('contract'). One significant difference from the treatment of derivational affixes should be noted. Inflectional affixes do not change the meanings and syntactic categories of the words to which they are attached to. To show their effects as grammatical morphemes, an attribute *mh* is added to the stem word units.

5.3 Discontinuous morphological phenomena

It should be noted that as far as affixes discussed above are concerned, we could also have them included under the < *wu* > tags of the stems they attach to. The more complicated treatment described above is in fact motivated by the discontinuous morphological phenomena as shown in the following examples, which are not attested in our corpus (not so much because of its small size, but because of its stylistic bias)

```
<pu>
<wu wi=1 gv=0 ct=vn mh="perf" sm="have meal">
  <du mu="2" tg=vnm sm="eat"><cu>"chi1"
<mu mi=1 wu=1><du tg=el><cu>"le"
<mu mi=2 wu=1><du tg=ncm sm="meal"><cu>"fan4"
</pu>
<pu>
<wu wi=1 gv=0 ct=adj mh="dup, de" sm="happy">
  <du mu="2" tg=ad><cu>"gao1"
<mu mi=1 wu=1><cu>"gao1"
<mu mi=2 wu=1><cu>"xing4"
<mu mi=3 wu=1><cu>"xing4"
<mu mi=4 wu=1><du tg=ed><cu>"de"
</pu>
```

In the first example, an infix *le* is inserted between the two characters of the word *chi1fan4*. As is common in computational linguistics research on Chinese, the segmentation program "segments" the text rather "lemmatize". It outputs three one-charter tokens, which, in Chinese, are all valid dictionary entries. The treatment of *le* ('PERF') is like *de*, which is to be expected. The constituent "word" *fan4* is separated from its "major" partner *chi1* in the compound word *chi1fan4* ('have meal'). It has to be marked as an < *mu* > attached to its major partner (as representative of the whole compound word) in order not to get into the way of subsequent syntactic analysis. The meaning of the < *wu* > is that of the compound word.

We "lemmatize", but we take care to make sure that the original output of the segmentation-tagging program is recoverable, just in case users of our annotated corpus are not happy with our lemmatization results.

The second example is even more interesting. The word is "inflected" form of the two-character dictionary entry *gao1xing4* ('happy'). The morphological process of reduplication has been applied, and each of the two characters is repeated to give a four-character surface form. As the segmentation-tagging program does not lemmatize and meddle with the order in which characters occur, its output is (somewhat erroneously) the four-token sequence of *gao1 gao1 xing4 xing4*. These characters are all valid dictionary entries in Chinese themselves, but they do not "combine" to form the word *gao1gao1xing4xing4*, which is obtained from *gao1xing4* by a kind of reduplication. In our annotation scheme, the four characters are one < *wu* > (with a number of < *mu* >'s attached to it. The original output of the segmentation program is preserved.

6 Problems with sharing

6.1 Incompatible syntactic structures

Syntactic structures produced according to different theoretical approaches are incompatible. Efforts like the PENN Treebank has tried to minimize the differences by adopting a basic bracketing scheme. But consider the following example from our corpus:

```
<pu>
<wu wi=1 gv=4 fn=mks ct=cnj sm="because">
  <du tg=jom><cu>"yin1"
<wu wi=2 gv=4 fn=sub ct=na sm="form">
  <du tg=nad><cu>"xing2"<cu>"shi4"
<wu wi=3 gv=4 fn=neg ct=adv sm="neg">
  <du tg=bu><cu>"bu4"
<wu wi-4 gv=8 fn=ajt ct=vt sm="conform">
  <du tg=vnm><cu>"he2"
<wu wi=5 gv=5 fn=obj ct=na
   sm="stipulation">
  <du tg=nad><cu>"gui1"<cu>"ge2"
<wu wi=6 gv=8 fn=mkm ct=cnj sm="then">
  <du tg=jom><cu>"er2"
```

```
<wu wi=7 gv=8 fn=neg cat=adv sm="neg">
  <du tg=bu><cu>"bu4"
<wu wi=8 gv=11 fn=ajt ct=aux sm="can">
  <du tg=um><cu>"neng2"
<wu wi=9 gv=8 fn=axo ct=vt sm="carry out">
  <du tg=vnd><cu>"li3"<cu>"xing2"
<wu wi=10 gv=8 fn=mka ct=de sm="rel">
  <du tg=ez><cu>"zhi1"
<wu wi=11 gv=0 fn=sub ct=nc sm="contract">
  <du tg=ncd><cu>"he2"<cu>"yue4"
</pu>
```

Linguists generally agree that the chunk from $< wu\ wi = 1 >$ to $< wu\ wi = 5 >$ is a "subordinate clause". but they may disagree about the internal structure of the clause. In GB/MP, the subordinating conjunction *yin1* ('because') will be the head of the tree hierarchy as shown below (unnecessary details skipped):

```
(yin1 (xing2shi4 bu4 he2 gui1ge2))
```

In HPSG, *yin1* may be analyzed as a "marker" or a "preposition" with a sentential complement. When analyzed as a marker, it will be a dependent of the verb *he2* ('conform')

```
((yin1) (xing2shi4) (bu4) he2 (gui1ge2))
```

When *yin1* is analyzed as a preposition, it may be considered the head of the phrase structure. However, it has been argued within HPSG that the preposition is a dependent of the verb *he2*.

The position of the subordinating conjunction in the syntactic structure of a subordinate clause is thus different depending on the syntactic theory followed. In our annotation, *yin1* is marked as a subordinating conjunction ($ct = cnj$). It is marked as being governed by the head word *he2*, for which it functions somewhat like a *marker* in HPSG (Pollard and Sag, 1994).

As we can obviously not claim impartiality for our analysis, our use of dependency annotation is of course not a solution to the problem. However, clearly indicating the dependency relationships in the parse structure will *accentuate* the disagreement. If the other researcher who wants to use our annotated corpus happens to favour an analysis that gives the same dependency relationships as we have marked in the annotation, then it will be up to him to flesh up the dependency structure with intermediate phrasal nodes according to his grammatical formalism. More often, the other researcher will find that the annotated syntactic structure does not agree completely with his own opinions, it should then be *easier* for him to make the necessary transformations if annotation consists only of skeletal dependencies without the complications arising intermediate nodes.

6.2 Shareable annotated corpora

It will be in vain for one to attempt to find parse structures that are universally accepted. What we can do, and have done, is to label arcs of our parse structures with the dependency relation names, thus leaving the hope alive that our parse structures may be convertible for use by researchers following other approaches.

While it will be out of our control whether other researchers will find our annotated corpora useful, we will be eager to be able to convert linguistic corpora annotated by other researchers for our own use.

Conclusion

We are giving SGML-based syntactic annotation to a small corpus of Chinese text as a piloting effort leading to large-scale syntactic annotation. We are also investigating the potential of importing and adapting annotated corpora prepared by researchers following other approaches. With the experience gained in this pilot effort, we will try to scale up to annotate a stylistically more balanced corpus. We also explore the possibility of making use of resources prepared by other researchers.

Acknowledgement

This research is supported by Grant No. 7000888, City University of Hong Kong.

References

Gosse Bouma, Rob Malouf, and Ivan A. Sag. 1998. A unified theory of complement, adjunct, and subject extraction. In Gosse Bouma, Geert-Jan M. Kruijff, and Richard T. Oehrle, editors, *Proceedings of the FHCG98. Symposium on Unbounded Dependencies*, pages 83–97, Saarbrücken. Universität des Saarlandes und DFKI.

Marie Bourdon, Lyne Da Sylva, Michel Gagnon, Alma Kharrat, Sonja Knoll, and Anna Maclachlan. 1998. A Case Study in Implementing Dependency-Based Grammars. In Alan Polguère and Sylvain Kahane, editors, *Proceedings of COLING-ACL'98 Workshop on Processing of Dependency-Based Grammars*, pages 88–94, Université de Montréal, August.

Joan W. Bresnan, editor. 1982. *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge, Massachusetts.

Noam Chomsky. 1986. *Barriers*. The MIT Press, Cambridge, Massachusetts.

Noam Chomsky. 1995. *The Minimalist Program*. The MIT Press, Cambridge, Massachusetts.

Jacques Courtin and Damien Genthial. 1998. Parsing with Dependency Relations and Robust Parsing. In Alan Polguère and Sylvain Kahane, editors, *Proceedings of COLING-ACL'98 Workshop* *on Processing of Dependency-Based Grammars*, pages 95–101, Université de Montréal, August.

Michael A. Covington. 1990. Parsing Discontinuous Constituents in Dependency Grammar. *Computational Linguistics*, 16(4):234–236.

Haim Gaifman. 1965. Dependency Systems and Phrase-Structure Systems. *Information and Control*, 8:304–337.

Eva Hajičova. 1991. Free Word Order Described Without Unnecessary Complexity. *Theoretical Linguistics*, 17:99–106.

David G. Hays. 1964. Dependency Theory: A Formalism and Some Observations. *Language*, 40:511–525.

Peter Hellwig. 1986. Dependency Unification Grammar. In *Proceedings of 11th International Conference on Computational Linguistics (COLING'86)*, pages 195–199.

Richard Hudson. 1984. *Word Grammar*. Blackwell, Oxford.

Nancy Ide, Greg Priest-Dorman, and Jean Véronis. 1996. Corpus Encoding Standard. Expert Advisory Group on Language Engineering Standards. Last modified 20 March 2000.

Tom Bong-Yeung Lai and Changning Huang. 1998a. An Approach to Dependency Grammar for Chinese. In Yang Gu, editor, *Studies in Chinese Linguistics*, pages 143–163. Linguistic Society of Hong Kong, Hong Kong.

Tom Bong Yeung Lai and Changning Huang. 1998b. Complements and Adjuncts in Dependency Grammar Parsing Emulated by a Constrained Context-Free Grammar. In Alan Polguère and Sylvain Kahane, editors, *Proceedings of COLING-ACL'98 Workshop on Processing of Dependency-Based Grammars*, pages 102–108, Université de Montréal, August.

Tom Bong Yeung Lai and Changning Huang. 1999a. Unification-based Parsing Using Annotated Dependency Rules. In Jost Gippert and Peter Olivier, editors, *Multilinguale Corpora: Codierung, Strukturierung, Analyse - 11. Jahrestagung der Gesellschaft für Linguistische Daten Verarbeitung (GLDV'99)*, pages 235–244. Enigma Corporation, Praha, December.

Tom Bong Yeung Lai and Changning Huang. 1999b. Unification-based Parsing Using Annotated Dependency Rules. In *Proceedings of 5th Natural Language Processing Pacific Rim Symposium 1999 NLPRS'99*, pages 102–107, Beijing, November.

T.B.Y. Lai, S.C. Lun, C.F. Sun, and M.S. Sun. 1992. A Tagging-Based First-Order Markov Model Approach to Automatic Word Identification for Chinese Sentences. In Julius T. Tou and Joseph J. Liang, editors, *Intelligent Systems for Processing Oriental Languages (Proceedings of the 1992 International Conference on Computer Processing of Chinese and Oriental Languages)*, pages 17–23, Tampa, Florida, December. Chinese Language Computer Society.

Tom B.Y. Lai, M.S. Sun, S.C. Lun, and B.K. Tsou. 1998. Using Syntactically Motivated Tags in Markov Model Word Segmentation. In *Proceedings of 1998 International Conference on Chinese Information Processing (ICCIP'98)*, pages 215–222, Beijing.

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

Carl Pollard and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Adam Przepiórkowski. 1999. On complements and adjuncts in Polish. In Robert D. Borsley and Adam Przepiórkowsk, editors, *Slavic in HPSG*, pages 183–210. CSLI Publications, Stanford.

Jane J. Robinson. 1970. Dependency Structures and Transformation Rules. *Language*, 46:259–285.

SGML. 1986. Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML). Genf: International Organization for Standardization. ISO8897.

Stanley Starosta. 1988. *The Case for Lexicase*. Pinter Publishers, London.

Lucien Tesnière. 1959. *Elements de syntaxe structurale*. Klincksieck, Paris.

The Italian Syntactic-Semantic Treebank: Architecture, Annotation, Tools and Evaluation

1] S. Montemagni, F. Barsotti, M. Battista, N. Calzolari, O. Corazzari, A. Zampolli
ILC-CNR / CPR, Pisa (Italy)
2] F. Fanciulli, M. Massetani, R. Raffaelli
Synthema, Pisa (Italy)
3] R. Basili, M.T. Pazienza, D. Saracino, F. Zanzotto
"Tor Vergata" University / CERTIA, Roma (Italy)
4] N. Mana, F. Pianesi
ITC-IRST, Trento (Italy)
5] R. Delmonte
Venezia University / CVR, Venezia (Italy)

Abstract

The paper reports on a multi-layered corpus of Italian, annotated at the syntactic and lexico-semantic levels, whose development is supported by a dedicated software augmented with an intelligent interface. The issue of evaluating this type of resource is also addressed.

Introduction

It is nowadays widely acknowledged that linguistically annotated corpora have a crucial theoretical as well as applicative role in Natural Language Processing. Italian still lacks such a resource. The paper describes a large scale effort to provide Italian with a multi-level annotated corpus, the Italian Syntactic-Semantic Treebank (henceforth referred to as ISST). Evaluation of ISST is foreseen in the framework of a machine translation application. Specifically developed software, including an intelligent interface, supports both annotation and evaluation activities.

ISST - which represents one of the main actions of an ongoing Italian national project, SI-TAL[1] - is developed by a consortium of companies and computational linguistics sites in Italy (see author's affiliations above). 1], 4] and 5] are in charge of the annotation, 3] of the design and construction of the annotation software and 2] of the evaluation of the developed resource.

Expected uses for ISST range from Natural Language Processing tasks (such as Information Retrieval, Word Sense Disambiguation, linguistic knowledge acquisition) to training (and/or tuning) of grammars and sense disambiguation systems, to the evaluation of language technology systems. ISST also promises to contribute to the start up of commercial systems for Italian processing. Last but not least, although annotated corpora are typically built and used in research and applicative contexts, their potential for teaching purposes has also to be emphasised; see, for instance, their use in the classroom for teaching syntax at Nijmegen University (Van Halteren 1997).

The final and tested version of ISST will be available in year 2001. Currently, the annotation phase is started, based on the linguistic guidelines and the annotation software which have just been released; yet, initial specifications remain subject to extensions and further refinements on the basis of feedback coming from the annotation process (e.g. emergence from the corpus of linguistic phenomena not yet covered by the specifications).

1 Architecture of ISST

ISST has a three-level structure ranging over syntactic and semantic levels of linguistic description. Syntactic annotation is distributed over two different levels, namely the constituent structure level and the functional relations level: constituent structure is annotated in terms of phrase structure trees reflecting the ordered

[1] SI-TAL is a joint enterprise leading towards an integrated suite of tools and resources for Italian Natural Language Processing, funded by the Italian Ministry of Science and Research (MURST) and coordinated by the Consorzio Pisa Ricerche (CPR).

arrangement of words and phrases within the sentence, whereas functional annotation provides a characterisation of the sentence in terms of grammatical functions (i.e. subject, object, etc.). The third level deals with lexico-semantic annotation, which is carried out here in terms of sense tagging augmented with other types of semantic information. The three annotation levels are independent of each other, and all refer to the same input, namely a morpho-syntactically annotated (i.e. pos-tagged) text which is linked to the orthographic file with the text and mark-up of macrotextual organisation (e.g. titles, subtitles, summary, body of article, paragraphs). The final resource will be available in XML coding.

The multi-level structure of ISST shows two main novelties with respect to other treebanks:

- it combines within the same resource syntactic and lexico-semantic annotations, thus creating the prerequisites for corpus-based investigations on the syntax-semantics interface (e.g. on the semantic types associated with functional positions of a given predicate, or on specific subcategorisation properties associated with a specific word sense);
- it adopts a distributed approach to syntactic annotation which presents several advantages with respect both to the representation of the syntactic properties of a language like Italian (e.g. its highly free constituent order) and to the compatibility with a wide range of approaches to syntax.

2 ISST input

2.1 Corpus composition

ISST corpus consists of about 300,000 word tokens reflecting contemporary language use. It includes two different sections: 1) a "balanced" corpus, testifying general language usage, for a total of about 210,000 tokens; 2) a specialised corpus, amounting to 90,000 tokens, with texts belonging to the financial domain.

The balanced corpus contains a selection of articles from different types of Italian texts, namely newspapers (*La Repubblica* and *Il Corriere della Sera*) and a number of different periodicals which were selected to cover a high variety of topics (politics, economy, culture, science, health, sport, leisure, etc.). The financial corpus includes articles taken from *Il Sole-24 Ore*. All in all, they cover a 10 year time period (1985-1995).

2.2 Morpho-syntactic annotation

Syntactic and lexico-semantic annotation takes as input the morpho-syntactically annotated text. Morpho-syntactic annotation was previously carried out at ILC in the framework of the European projects PAROLE (Goggi et al. 1997) and ELSNET (Monachini and Corazzari 1995). The text was automatically tagged; the output was manually revised by a team of linguists. The adopted morpho-syntactic tagset conforms to the EAGLES international standard (Monachini and Calzolari 1996).

Annotation at this level involves identification of morphological words with specification of part of speech, lemma, and morho-syntactic features such as number, person, gender, etc.

Morphological words typically stand in a one-to-one relation with orthographic words with two exceptions, namely: i) the case of more than one morphological word which forms part of the same orthographic word (as in the case of cliticized words, e.g. *dammelo* 'give+to_me+it'); ii) the case of more than one orthographic word which make up a single morphological word not otherwise decomposable (as in the case of multi-word expressions such as *ad_hoc*, *al_di_là* 'beyond', *fino_a* 'up_to').

3 ISST annotation schemata

3.1 General requirements

The design of each individual annotation schema underlying ISST and their interrelations are intended to fit a list of basic requirements following directly from the typology of foreseen uses. They include:

a) usability in both real applications and research purposes;

b) compatibility with different approaches to syntax, both dependency- and constituency-based, either adopted in theoretical or applicative frameworks;

c) applicability on a wide scale, in a coherent and replicable way;

d) applicability to both written and spoken language (this requirement does not apply to the actual ISST but it is foreseen in view of possible resource extensions to spoken language data).

Within ISST, requirements a) and b) are satisfied by distributing the annotation over different levels (mainly for what concerns syntactic annotation) and, for each level, by factoring out different information types according to different dimensions.

Different strategies are pursued to meet requirement c). This is achieved at the level of individual annotation schemes by first providing wide coverage and detailed annotation criteria and then by avoiding as much as possible arbitrary annotation decisions (i.e. uncertainty cases are preferably dealt with through underspecification or disjunction over different interpretations). c) has also consequences on the relationship between different annotation levels: redundancy is avoided as much as possible; i.e. a given information type has to be specified only once, at the relevant annotation level (e.g. grammatical relations such as subject and object are only specified at the functional level).

Finally, d) is guaranteed by the independence of syntactic annotation levels: spoken data, which are typically fraught with ellipses, anacolutha, syntactic incompleteness and other related disfluency phenomena cannot be easily represented in terms of constituency. By contrast, the level of functional analysis - which in ISST has an independent status - naturally reflects a somewhat standardised representation, since it abstracts away from the surface realisation of syntactic units in a sentence, thus being relatively independent of disfluency phenomena and incomplete phrases.

3.2 Syntactic annotation

Most treebanks, currently available or under construction for different languages, adopt a unique syntactic representation layer, following either a costituency-based approach (see, among many others, Marcus et al. 1993, Sampson 1995, Greenbaum 1996, Sandoval et al. 1999) or a dependency-based one (e.g. Karlsson et al. 1995), or a hybrid one combining features of both (e.g. Brants et al. 1999, Abeillé et al. 2000). ISST departs from all of them since it adopts a multi-level structure.

To our knowledge, the only multi-level treebank is the Prague Dependency Treebank (PTD, Bémová et al. 1999), but in this case the different annotation levels refer respectively to a) the surface dependency relations and b) the underlying sentence structure. By contrast, ISST adopts a monostratal view of syntax, and thus both syntactic annotation levels are rather intended to provide orthogonal views of the same surface syntax. These views, though complementary, are developed independently of each other.

This bi-level approach to syntactic representation is particularly suited to deal with a language like Italian, which allows for considerable variation in the ordering of constituents at the sentence level. In fact, by decoupling functional information from the constituent structure, the treatment of word order variation does not interfere in any way with the representation of functional relations, i.e. the encoding of the latter becomes entirely separate from the order of contituents in the sentence.

3.2.1 Constituency annotation

In ISST, constituency annotation departs from other constituency-based syntactic annotation schemes (e.g. the one adopted in the Penn Treebank) in a number of respects, due to: a) the peculiarities of Italian as a free constituent order language; b) the distributed organisation of syntactic annotation in ISST.

Constituency annotation in ISST uses an inventory of 22 constituent types (see table below). Specialized constituent names are used for a number of complements or adjuncts, in order to help the mapping with functional annotation.

Const type	**Meaning**	**Classif**
F	sentence	structural
SN	noun phrase, including its complements and/or adjuncts	substantial
SA	adjectival phrase, including its complements and/or adjuncts	lexical
SP	prepositional phrase	lexical
SPD	prepositional phrase *di* 'of'	lexical
SPDA	prepositional phrase *da* 'by, from'	lexical
SAVV	adverbial phrase, including its complements and/or adjuncts	substantial
IBAR	verbal nucleus with finite tense and all adjoined elements like clitics, adverbs and negation	substantial
SV2	infinitival clause	substantial
SV3	participial clause	substantial
SV5	gerundive clause	substantial

Const type	Meaning	Classif
FAC	sentential complement	lexical
FC	Coordinate sentence (also ellipsed and gapped)	lexical
FS	Subordinate sentence	lexical
FINT	+wh interrogative sentence	lexical
FP	punctuation marked, parenthetical or appositional sentence	lexical
F2	relative clause	lexical
CP	dislocated or fronted sentential adjuncts	structural
COORD	Coordination with coordinating conjunction as head	lexical
COMPT	Transitive/Passive/Ergative/Reflexive Complement	structural
COMPIN	Intransitive/Unaccusative Complement	structural
COMPC	Copulative/Predicative Complement	structural

From the point of view of their relations to functional labelling, syntactic constituents are divided up into two main subgroups (see column 3 of the table above): functional constituents and substantial constituents. This subdivision reflects theoretical assumptions which are derived from the Lexical Functional Grammar theory. In particular, functional constituents are internally subdivided into structural constituents (used to set complements apart) and lexical constituents (headed by a lexical head with or without semantic content). Structural constituents also contain F and CP where F has the task of indicating the canonical sentential constituent and CP indicates the presence of sentential adjuncts, or some discontinuity in the utterance.

At the same time, the fact that in ISST functional relations are dealt with at a distinct level instead of being defined in terms of constituent structures allows ISST to dispense with empty elements such as null subjects or traces, thus making annotation more intelligible. In fact, the relevant information is recovered at the functional level, through a relation linking the displaced element to its head. Therefore, syntactic phenomena such as pro-drop, ellipsis as well as cases of discontinuous or non canonical order of constituents (topicalisation, wh-questions, etc.) are not accounted for in terms of empty categories and coindexation as e.g. in the Penn Treebank but rather at the functional annotation level.

Constituency annotation of ISST is worked out in a semi-automatic way. First, the text is parsed by a Shallow Parser (Delmonte 1999, 2000) whose task is that of building shallow syntactic structures for each safely recognizable constituent. In uncertainty cases, no attachment is performed at this stage in order to avoid being committed to structural decisions which might then reveal themselves to be wrong. In fact, it is preferable to perform some readjustment operations after structures have been built rather than introducing errors from the start. Then, the output of the shallow parser is manually revised and corrected.

3.2.2 Functional annotation

Functional annotation in ISST is carried out by marking relations between words belonging to major lexical classes only (i.e. non-auxiliary verbs, nouns, adjectives and adverbs), independently of previous identification of phrasal constituents. Advantages of this choice include, on the theoretical front, the fact that ISST can be used as a reference resource for a wider variety of different annotation schemes, both constituency- and dependency-based ones (Lin 1998). Moreover, on the applicative side, head-based functional annotation is comparatively easy and "fair" to be used for parsing evaluation since it overcomes some of the well-known shortcomings of constituency-based evaluation (see, among others, Carroll et al. 1998, Sampson 1998, Lin 1998). Last but not least, head-based functional annotation is naturally i) multi-lingual, as functional relations probably represent the most significant level of syntactic analysis at which cross-language comparability makes sense, and ii) multi-modal, since it permits comparable annotation of both spoken and written language.

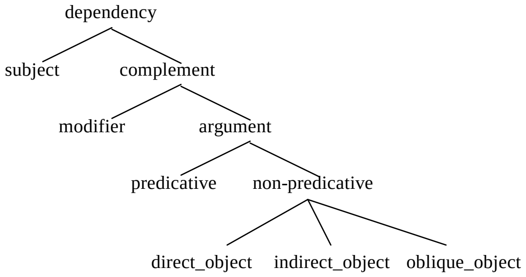
FAME (Lenci et al. 1999, 2000) is the annotation scheme (originally developed in the SPARKLE project LE-2111 and then revised in the framework of ELSE LE4-8340) adopted for functional annotation in ISST, which has been revised and integrated to make it suitable for annotation of unrestricted Italian texts. The building blocks of FAME are functional relations, further subdivided into dependency relations and other relation types dealing with coordination phenomena and clause-internal co-referential bonds. Only the former are described below for sake of paper length.

A dependency relation is an asymmetric binary relation between full words, respectively a head and a dependent. Each dependency relation is expressed as follows:

```
dep_type (lex_head.<head_features>,
          dependent.<dep_features>)
```

where dep_type specifies the relationship holding between the lexical head and its dependent. At this level, either the head or the dependent can correspond to elliptical material; this makes it possible to deal with pro-drop phenomena and other types of elliptical constructions.

Dep_types are hierarchically structured to make provision for underspecified representations of highly ambiguous functional analyses (see above). The typology of dependency relations, hierarchically organized, is given below.



In the proposed scheme a crucial role is played by the features associated with the elements of the relation, which complement relational information. Features convey, for instance, information about the grammatical word (preposition or conjunction) which possibly introduces the dependent in a given relation, or about the open/closed predicative function of clausal dependents (in this way control information is also encoded).

Functional annotation in ISST is thus modularly represented, i.e. it is structured into relational and feature information, each factoring out different but interrelated facets of functional annotation. This modular representation provides the prerequisites for ISST to be used as a reference annotation scheme which is compatible with a wide range of theories and thus mappable onto different syntactic representation formats (for more details on the intertranslatability of FAME into other syntactic representation formats see Lenci et al. 1999, 2000).

Annotation at the functional level is carried out manually.

3.2.3 *Annotation examples*

The sketchy description of the syntactic annotation schemes provided above is complemented here with annotation examples. The two ISST syntactic annotation levels, the constituent structure and the functional ones, are developed independently; in spite of this fact, they are strictly interrelated and complement each other.

In order to show the peculiarities of the two annotation levels and their interrelations, let us consider the ISST annotation of the following Italian sentence, *Giovanni sembra arrivare domani* 'John seems to arrive tomorrow':

- **Constituent structure annotation**

```
f-[    sn-[Giovanni],
       ibar-[sembra],
       sv2-[arrivare,
              savv-[domani]]]
```

- **Functional annotation**

```
sogg (sembrare, Giovanni)
arg (sembrare,
       arrivare.<status= aperto>)
mod (arrivare, domani)
sogg (arrivare, Giovanni)
```

Note that the subject relation holding between *arrivare* and *Giovanni* in the functional annotation does not find an explicit counterpart at the level of constituent structure representation since subject raising is not treated at that level.

Depending on the expected uses, the two annotation layers can be accessed and examined independently. However, due to the complementarity of the information contained in them, combined views on the developed resource can also be obtained. For instance, projection of functional information onto the constituent structure results as follows:

```
f-[    sn-sogg[Giovanni],
       ibar-[sembra],
       sv2-arg[arrivare,
                savv-mod[domani]]]
```

where each constituent category is marked, whenever possible, with a functional tag. This is one of the many possible combined views which can be obtained on the ISST syntactically annotated corpus.

3.3 Lexico-semantic annotation

3.3.1 Basics

The strategy set-up for annotation at this level takes advantage of two previous experiments of semantic tagging carried out at ILC in the framework of the SENSEVAL initiative (Calzolari et al., forthcoming) and of the ELSNET resources task group activity (Corazzari et al., 2000).

In ISST, lexico-semantic annotation consists in the assignment of semantic tags, expressed in terms of attribute/value pairs, to full words or sequences of words corresponding to a single unit of sense (e.g. compounds, idioms). In particular, annotation is restricted to nouns, verbs and adjectives and corresponding multi-word expressions.

ISST semantic tags convey three different types of information:

1) sense of the target word(s) in the specific context: ItalWordNet (henceforth, IWN) is the reference lexical resource used for the sense tagging task (CPR et al., 2000). IWN, developed from the EuroWordNet lexicon (Alonge et al. 1998), includes two parts, a general one and a specialized one with financial and computational terminology;
2) other types of lexico-semantic information not included in the reference lexical resource, e.g. for marking of figurative uses;
3) information about the tagging operation, mainly notes by the human annotator about problematic annotation cases.

Note that through the taxonomical organisation of IWN word senses an implicit assignment is made to the semantic types of the IWN ontology. In this way, ISST sense tagging can also be seen as semantic tagging.

Starting from the assumption that senses do not always correspond to single lexical items, the following typology of annotation units is identified and distinguished in ISST:

us: sense units corresponding to single lexical items (either nouns, verbs or adjectives);

usc: semantically complex units expressed in terms of multi-word expressions (e.g. compounds, support verb constructions, idioms);

ust: title sense units corresponding to titles of any type (of newspapers, books, shows, etc.). Titles receive a two-level annotation: at the level of individual components and as a single title unit.

3.3.2 Annotation criteria

Each annotation unit is tagged with the relevant sense according to IWN sense distinctions. In order to meet requirement c) in section 3.1 above, arbitrary sense assignments, which may occur when more than one IWN sense applies to the context being tagged, are avoided by means of underspecification (expressed in terms of disjunction/conjunction over different IWN senses).

The other lexico-semantic tags allow to mark:

- a us or usc used in a metaphoric or methonymic or more generally in a figurative sense: e.g. *la molla di una simile violenza* 'the spring of such a violence' where *molla* is used in a metaphoric sense. The distinction between lexicalized and non lexicalized figurative usages can be inferred from the assigned IWN sense: non lexicalized figurative uses are linked to the literal sense;
- a us semantically modified through evaluative suffixation (e.g. *appartamentino* 'small flat', *concertone* 'big concert');
- the semantic type (i.e. human entity, artifact, institution, location, etc.) of proper nouns, either us (e.g. *pds* 'the pds party' is semantically tagged as a 'group') or usc (e.g. *Corno d'Africa* 'the Horn of Africa' is assigned the sematic type of 'place');
- the usc subtype, e.g. compound (e.g. *prestito obbligazionario* 'loan stock'), idiom (e.g. *mettere i puntini sulle i* 'to dot one's i's'), support verb construction (e.g. *dare aiuto* 'to give assistance');
- the ust subtype, i.e. title of an opera (e.g. *Il barbiere di Siviglia*), of a newspaper (e.g. *La Nazione*) or of something else.

In this way, the annotated corpus provides more than a list of instantiations of the senses attested in the reference lexical resource. Through the added value of this additional information, the annotated corpus becomes a repository of interesting semantic information going from titles and proper nouns to non-lexicalized metaphors, metonymies and evaluative suffixation, and in general to non-conventional uses of a word.

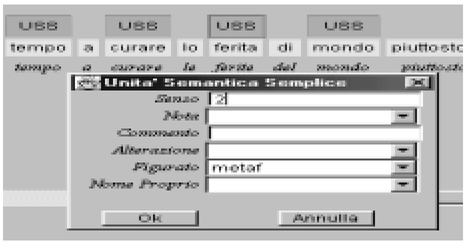
Finally, notes about the tagging operation are mainly used to ease and speed up the annotation process and its revision: the human annotator can keep track of problematic cases (e.g. cases of indistinguishable IWN senses, of ambiguous corpus contexts, etc.). Input of this type may also be useful for discussion with the team of IWN lexicographers with a view to prospective revisions and updating of the lexical resource.

As to the annotation methodology for this level, in order to ensure that polysemous words and usc are tagged consistently, the annotation is manually performed ‘per lemma’ and not sequentially, that is, word by word following the text.

3.3.3 Annotation examples

Let us exemplify the annotation strategy illustrated in the previous sections with a few semantically tagged corpus occurrences.

An example of an annotated us is given below: the target word is *ferite* 'wounds' in the context *curare le ferite del mondo* 'to cure the wounds of the world'. In the annotation window, the target word is assigned the sense number 2; the feature *figurato=metaf* marks its metaphoric use in the specific corpus context.



Unita' Semantica Semplice

Senso: 2

Nota:

Commento:

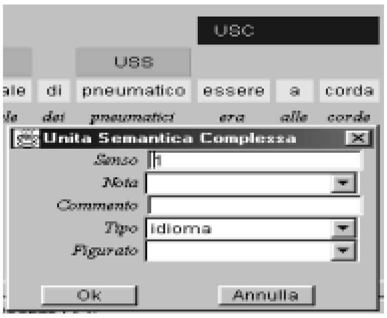
Alterazione:

Figurato: metaf

Nome Proprio:

USS
USS
USS
USS
tempo a curare lo ferita di mondo piuttosto
tempo a curare le ferite del mondo piuttosto
Ok
Annulla

Annotation of semantically complex units (usc) is exemplified below for the multi-word expression *essere alle corde* 'to be hard-pressed':



Unita Semantica Complessa

Senso: 1

Nota:

Commento:

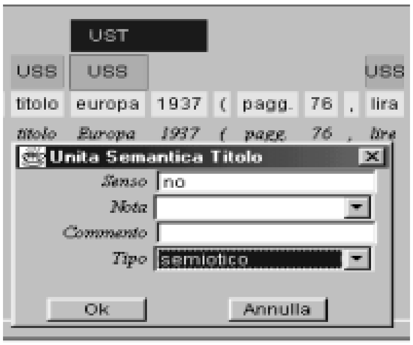
Tipo: idioma

Figurato:

USC
USS
ale di pneumatico essere a corda
le dei pneumatici era alle corde
Ok
Annulla

The blue box covering the text shows that it has been marked as a usc; the annotation window specifies its sense number (1) in IWN and its type (idiom).

Finally, an example is given below for title sense units, or ust. It can be noticed that the book title *Europa 1937* 'Europe 1937' is annotated both at the level of its constituting words (see *Europa*) and as a single unit of type title of a book (*tipo=semiotico*). Obviously, sense information does not apply to ust.



Unita Semantica Titolo

Senso: no

Nota:

Commento:

Tipo: semiotico

UST
USS
USS
USS
titolo europa 1937 (pagg. 76 , lira
titolo Europa 1937 (pagg. 76 , lire
Ok
Annulla

4 The multi-level Linguistic Annotation Tool

The labour intensive annotation task demands for tools devoted to access efficiently the large amount of textual data and the related annotations. In this perspective, both a data model and effective graphical representations are mandatory.

GesTALt is the annotation tool defined for ISST where an object oriented data definition has been preferred for its flexibility. Specific data models and graphical representations are defined so to comply with the different needs of the three levels of annotation. Building upon these data models, level-oriented subsystems are settled. The tool is also designed to ease the control of intra-level and inter-levels coherence.

4.1 The linguistic data base

The model of linguistic data is designed within the object oriented formalism. The defined data are directly used in the object oriented database underlying *GesTALt*. For each level of annotation, a specific container has to be defined. The system (and its subsystems) manages a collection of documents, the corpus: this relation is represented in a class hierarchy.

Moreover, the different level interpretations associated with sentences in the corpus are modeled respectively via the class of objects. To give the flavor of the object modeling of linguistic structures, we present here the hierarchy describing constituent annotation (i.e. the class *synt_int*).

Constituency annotation is based on tree structures where both internal nodes and leaves are constituents (*const*). Leaves are called *basic constituents* (*b_const*), while internal nodes *complex constituents* (*c_const*). The resultant *synt_int* sub-hierarchy is depicted in Fig. 1.

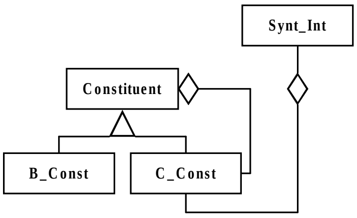


Fig. 1 Syntactic interpretation

Complex constituents are collections of constituents, either basic or complex ones. A constituency-based syntactic interpretation is thus the complex constituent representing the interpretation of the whole sentence. This notion is modeled by the relation between the *c_const* class and the *synt_int* class in the hierarchy.

4.2 The visual representation of annotation

Managing large sentence annotations is cumbersome. Effective graphical representations are needed both for the annotator and the user. Their aim is to ease the navigation in intricate information.

Constituency-based annotation schemes are tree structures. Graphical tree representations aim to ease the user interactions with the tree structures, i.e. the display, retrieval and updating of annotation.

The visual representation defined is a *strip tree* (see Fig. 2) which resembles standard bracketed representations and provides an intuitive and easy to modify hierarchical view of the constituent structure.

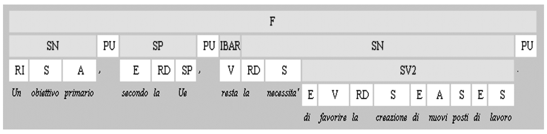


Fig. 2 Strip tree

Functional annotation is visually represented in terms of graphs, where functional relations are drawn as arcs linking the head and the dependent. The insertion/deletion of elliptical material is another essential feature of this tool module.

Finally, lexico-semantic annotation, which proceeds per lemma, does not pose specific representation requirements, while browsing at this level needs the parallel use of the IWN tool.

4.3 *GesTALt* architecture

The *GesTALt* annotation workbench is the resultant system, constituted by a pool of cooperative subsystems. The system manages the linguistic database sketched in section 4.1 and allows its output in the XML standard.

The system is a suite composed by specific applications: *SinTAS* for constituent annotation; *FunTAS* for functional annotation; *SemTAS* for lexico-semantic annotation; and *ValTAS* for evaluation and correction of inter- and intra-level annotations.

FunTAS, *SinTAS*, and *SemTAS* are stand alone applications. The synthesis of the three subsystems is obtained in *ValTAS* that need all the capabilities spread in the subsystems. The technologies adopted for the development (object-oriented design), in conjunction with an ad-hoc architectural design, allows an easy reuse of the functionalities developed for the subsystems in the global (i.e. *ValTAS*) system.

The overall *GesTALt* architecture is shown in Fig. 3 (overleaf), where components are represented as boxes, and interactions as arrows.

The creating/translating flow of the object-oriented database (*GestTALt–OODB*) is shared by the subsystems. Information is extracted from and injected in XML containers via specific wrappers (*Wrapper-in* and *Wrapper-out*) . The *GestTALt–OODB* is the object oriented database where the annotation of the different levels is stored respectively by *FunTAS*, *SinTAS* and *SemTAS*, together with the morphologically annotated corpus used as input by all annotation modules.

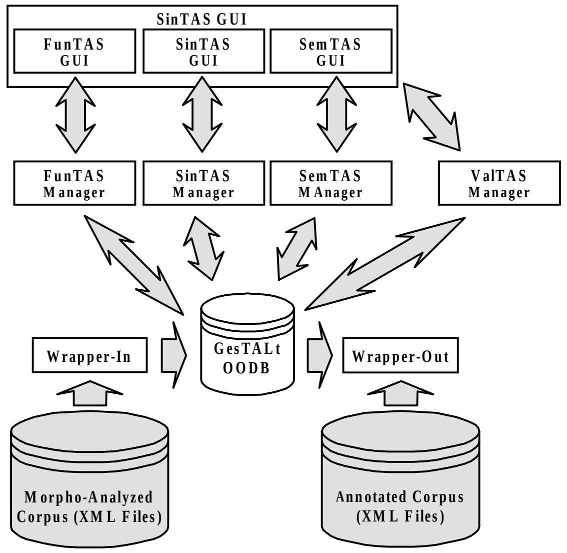


Fig. 3 GesTALt architecture

Each subsystem, but *ValTAS* that include all, is composed by specialized components. The graphical user interfaces based on the specific representations are depicted in the general architecture (*FunTAS GUI*, *SinTAS GUI*, *SemTAS GUI* and *ValTAS GUI*, respectively). Furthermore, the different ways of interaction with the database impose the design of special modules devoted to ad-hoc navigation of the hierarchy (*FunTAS Manager*, *SinTAS Manager*, *SemTAS Manager*, and *ValTAS Manager*).

5 Treebank Evaluation

The information stored in ISST, in particular in the financial corpus, will be used to improve an automatic Italian-English translation system, PeTra Word 2.0 , developed by Synthema and already on the market.

PeTra is based on the Logical Grammars ("Slot Grammars") formalism (McCord 1980, 1989) and is composed of three main components: the Italian language analyser (morphologic analyser, monolingual dictionary and syntactic parser), the transfer component (bilingual dictionary and structural transfer rules) and the English morphologic generator. We expect to improve: dictionaries, Italian grammar and transfer rules.

5.1 Changes to the dictionary content

Adding the missing entries: PeTra's dictionary coverage will be enlarged through addition of missing specialised entries and through improvement of already contained entries. Associated translations will be added to the bilingual dictionary.

Inserting new multi-word expressions: the multi-word expressions annotated in ISST will be revised and added to the dictionary either in terms of single entries or of particular constructions associated with component words, considering the system constrains.

Improving lexico-semantic hierarchy: by using lexico-semantic annotation, the semantic-hierarchical dictionary structure will be revised: the semantic attributes are especially used for the lexical transfer disambiguation.

5.2 Analysis Rules

The current grammar has a good coverage (i.e. 88% on unrestricted texts), but it is likely that many structures in the ISST corpus will be analysed incompletely or incorrectly: the corpus is a specialised one and it may contain constructions which are not used in standard Italian. ISST will be examined to check the grammar coverage: accessing ISST on the basis of functional relations, which correspond to the slots, will allow to study the features and the Constituents, in order to determine the possible structures and encode the proper rules.

The translation tests will also allow to determine the sentences which are not recognised by the current grammar: the rules will be modified by retrieving the “similar” structures contained into ISST. The access to ISST will be made through the sentence being examined in order to obtain the two syntactic annotations, study them to determine the uncovered structure and other possible annotations of the same type inside the corpus, and finally analyse them to decide whether and how to apply possible changes.

5.3 Transfer Rules

By analysing all of the new elements included into the analysis rules and revising the translation tests, the set of rules which forms the syntactic transfer can be improved.

5.4 Results Evaluation

The result validation will be made by comparing the translations of texts in the ISST financial corpus. These translations will be obtained before and after the system tuning. The evaluation will verify the improvement obtained.

The software, which will be a support product for the evaluator, will allow to interactively access to the source text and the related translations, and assign a score based on fixed criteria. The evaluation system will also automatically evaluate the amount of unknown words and not closed trees.

References

Abeillé A., Clément L., Kinyon A (2000) *Building a treebank for French*, in Proceedings of LREC-2000, 31/5-2/6 2000, Athens, pp. 87-94.

Alonge, A., Calzolari, N., Vossen, P., Bloksma, L., Castellon, I., Martì, T., Peters, W. (1998) *The Linguistic Design of the EuroWordNet Database* in Special Issue on EuroWordNet. Computers and the Humanities, 32, 2-3, pp. 91-115.

Brants T., Skut W., Uszkoreit H. (1999) *Syntactic annotation of a German newspaper corpus*, in Proceedings of the Treebanks workshop, Journée(s) ATALA sur les corpus annotés pour la syntaxe, 18-19 juin 1999, Université Paris 7, Paris.

Bémová A., J. Hajic, B. Hladká, J. Panenová (1999) *Syntactic tagging of the The Prague dependency Treebank*, in *Proceedings of the Treebanks workshop, Journée(s) ATALA sur les corpus annotés pour la syntaxe*, 18-19 juin 1999, Université Paris 7, place Jussieu, Paris.

Calzolari N. and Corazzari O. (forthcoming) *Senseval/Romanseval: the framework for Italian*. Computers and the Humanities, Dordrecht.

Carroll J., E. Briscoe, A. Sanfilippo (1998) *Parser Evaluation: a Survey and a New Proposal*, in LREC-1998 Proceedings, Granada, Spain, 28-30 May, pp. 447-454.

Corazzari O., Calzolari N., Zampolli A. (2000) An Experiment of Lexical-Semantic Tagging of an Italian Corpus. LREC-2000 Proceedings, Athens.

Corazzari O., M. Monachini, 1995, *ELSNET: Italian Corpus Sample*, ILC-CNR, Pisa.

CPR, ITC-irst, Quinary (2000) ItalWordNet: Rete semantico-lessicale per l'italiano. SI-TAL, Specifiche Tecniche di SI-TAL, Manuale Operativo, Capitolo 2.

Delmonte R. (1999), *From Shallow Parsing to Functional Structure*, in Atti del Workshop AI*IA "Elaborazione del Linguaggio e Riconoscimento del Parlato", IRST Trento, pp.8-19.

Delmonte R. (2000), *Shallow Parsing And Functional Structure In Italian Corpora*, LREC-2000 Proceedings, Athens, June 2000.

Goggi S., L. Biagini, E. Picchi, R. Bindi, S. Rossi, R. Marinelli (1997) *Italian Corpus Documentation*, LE-PAROLE WP2.11, ILC, Pisa.

Greenbaum S. (ed.) (1996) *English Worldwide: The International Corpus of English,* Oxford, Clarendon Press.

Van Halteren H. (1997) *Excursions into syntactic databases*, Amsterdam, Rodopi.

Karlsson F., Voutilainen A., Heikkila J., Anttila A. (eds.) (1995) *Constraint Grammar, a language-independent system for parsing unconstrained text.* Berlin e New York: Mouton de Gruyter.

Lenci A., S. Montemagni, V. Pirrelli, C. Soria (1999) *FAME: a Functional Annotation Meta-scheme for Multimodal and Multi-lingual Parsing Evaluation*, Proceeding of the ACL99 Workshop on Computer-Mediated Language Assessment and Evaluation in NLP, University of Maryland, June 22nd .

Lenci A., S. Montemagni, V. Pirrelli, C. Soria (2000) *Where opposites meet. A Syntactic Meta-scheme for Corpus Annotation and Parsing Evaluation*, LREC-2000 Proceedings, Athens, June 2000.

Lin D. (1998) *A dependency.based method for evaluating broad-coverage parsers*, Natural Language Engineering 4(2), pp. 97-114.

Marcus M., Marcinkiewicz M.A., Cantorini B. (1993) *Building a Large Annotated Corpus of English: The Penn Treebank*, Computational Linguistics, 19(2), pp. 313-330.

McCord M.C. (1980) *Slot Grammars.* Computational Linguistics, vol 6, pp 31-43.

McCord M.C. (1989) *Design of LMT: A Prolog-based Machine Translation System* Computational Linguistics, vol 15, pp. 33-52.

Monachini M., Calzolari N. (1996) *Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicons and Corpora. A Common Proposal and Application to European Languages.* EAGLES Recommendations. Pisa, ILC.

Sampson G. (1995) *English for the Computer*, Oxford, Clarendon Press.

Sandoval M., Lopez Ruesga A., Sanchez León S. and F. (1999) *Spanish Tree Bank: Specifications*, Version 4, Manuscript.

SI-TAL (2000) Specifiche Tecniche di SI-TAL. Manuale Operativo.

Where Should Annotation Stop?

Geoffrey Sampson
University of Sussex

ABSTRACT

The paper asks how much structural detail it is reasonable to include in a detailed general-purpose grammar annotation scheme. I argue that there is no principled answer to that question; even grammatical distinctions which in general are clear and linguistically central will often be "distinctions without a difference" in particular examples. The discipline which offers the closest intellectual precedent for linguistic treebank-compilation activity, biological systematics, is disanalogous from our work in that respect.*

Detailed v. skeleton analytic schemes

Any scheme for structural annotation of corpora must embody decisions about how much detail to include.

Some groups explicitly aim at "skeleton parsing", marking much less grammatical detail than linguists recognize a language as containing. In many circumstances, this will be a sensible strategy. If one's chief goal is to have as large as possible a quantity of analysed material, from which reliable statistics can be derived, then skeleton parsing is more or less unavoidable. Automatic parsers may be able to deliver skeleton but not detailed analyses, and human analysts can produce skeleton analyses quickly. Furthermore, for some natural language processing applications skeleton analysis may be all that is needed.

But attention also need to be given to detailed structural analysis. All the grammar in a language, surely, serves some function or another for users of the language – it is not just meaningless ornamentation. There are many diverse potential applications for automatic NLP, some of which have scarcely begun to be developed, and it would be rash to assume that this or that aspect of language structure can safely be ignored because it will never be relevant for any practical NLP application. If some minor details of structure might be significant for research in the future, then the sooner we begin devising standardized, explicit ways of registering them in our treebanks (structurally analysed corpora) the better, because the business of evolving usable, consistent schemes of structural classification and annotation is itself a challenging and time-consuming activity.

To draw an analogy from the biological domain, much of the range of very lively research developments currently taking place in genetics and cladistics depends on the fact that biologists have a detailed, internationally-recognized system for identifying living species, the foundations of which were laid down as long ago as the eighteenth century. Linnaeus and his successors could not have guessed at the kinds of research revolving round DNA sequences which are happening in biology nowadays, but modern biology would be hampered if their species-identification scheme were not available.

Since the 1980s, my team has been developing a structural annotation scheme for English (a first draft of which was published as Sampson (1995))

* This research was supported by the Economic and Social Research Council (UK).

which aims at rigorous explicitness and maximum completeness of detail. We have also been compiling and circulating treebanks which apply the scheme to language samples, but the level of detail of the analytic scheme means that the treebanks illustrating it are small compared to some of those nowadays available – we accept this as a necessary cost of our strategy. To quote the documentation file of our SUSANNE Corpus (`ftp://ota.ox.ac.uk/pub/ota/public/ susanne/`):

> The SUSANNE scheme attempts to provide a method of representing all aspects of English grammar which are sufficiently definite to be susceptible of formal annotation, with the categories and boundaries between categories specified in sufficient detail that, ideally, two analysts independently annotating the same text and referring to the same scheme must produce the same structural analysis.

Comprehensiveness and rigour of analytic guidelines are ideals which can never be perfectly attained, but there is some evidence that the SUSANNE scheme is recognized as having made a useful advance; for instance, Terence Langendoen (President of the Linguistic Society of America) commented in a review that its "detail … is unrivalled" (Langendoen 1997: 600).

If one's aim is a comprehensive detailed rather than skeleton analytic scheme, then a question which arises and which does not seem to have been much discussed to date is where to stop. How does one decide that one has exhausted the range of grammatical features which are "sufficiently definite to be susceptible of formal annotation"?

The trainability criterion

In practice, one factor that may impose limits on detail is what it is practical to teach annotators to mark reliably. Even if an annotation scheme is limited to standard, traditional grammatical categories, it is hard to overestimate the difficulty of training research assistants to apply it to real-life language samples in a consistent manner. Some annotation projects are explicit about ways in which training considerations shaped their notation scheme. Meteer et al. (1995), defining the dysfluency annotation scheme of the Switchboard Corpus, make remarks such as "annotators were basically unable to distinguish the discourse marker from the conjunctive use of *so*", "*actually* also proved impossible for the annotators to mark consistently and was jettisoned as a discourse marker part of the way through".

But, although what one can and cannot train annotators to do is obviously an important consideration in practice, it is hard to accept it as a principled boundary to detail of annotation. Sometimes, annotators' failure to apply a distinction consistently may be telling us that the distinction is unreal or inherently vague. But there are certainly other cases where the distinction is real enough, and annotators are just not good at learning it (or a principal investigator is not good at teaching it). Usually, leaders of annotation projects are senior and more linguistically experienced than the annotators employed by the projects, so taking trainability as decisive would mean systematically ascribing more intellectual authority to the inexpert than to the expert.

Limits to expert decision-making

In principle, what junior annotators can learn to do is a secondary consideration, which is likely to depend on factors such as time available for training and individual educational background, as much as on the properties of the language itself. More scientifically interesting is the fact that sometimes it seems difficult or impossible to devise guidelines that enable even linguistic experts to classify real-life cases consistently.

If some grammatical distinction is hard for an expert to draw in a majority of cases, then probably we would all agree that that distinction is best left out of our annotation scheme. An example might be the distinction, among cases of the English pronoun *they*, between the original use referring to plural referents, and the newer use, encouraged recently in connexion with the "political correctness" movement, for a singular referent of unknown sex. This probably deserves to be called a grammatical distinction; note for instance that "singular *they*" forms a reflexive as *themself* rather than *themselves*, as in the following British National Corpus examples:

> *Critics may claim inconsistency, but the person involved may justify themself by claiming total consistency*. FA9.01713

> *… the person who's trying not to drink so much and beats themself up when they slip back and get drunk!* CDK.02462

(These are not isolated oddities; traditionalists may be surprised to learn that 23 of the 4124 BNC texts each contain one or more tokens of the form *themself*, which seems quite a high number considering that singular *they* is unquestionably far less frequent than plural *they*.) But (although I have not checked this) it seems likely that in a high proportion of cases where *they* is in fact being used for "he or she", there will be few or no contextual cues to demonstrate that it is not used with plural reference – sometimes even for the speaker or writer it may be intended as nonspecific with respect to number as well as sex. So I would not want to add a distinction between singular and plural *they* to our annotation scheme, and I imagine few colleagues would advocate this for general-purpose linguistic annotation schemes. (If an annotation scheme is devised for some specialized purpose, there is obviously no saying what distinctions it may need to incorporate.)

More problematic are the many grammatical distinctions which can often be made easily, and which may seem to the linguistic expert (and perhaps to less expert annotators) rather basic to the structure of the language, but which in particular cases may be hard to draw. What proportion of instances of a distinction need to be indeterminate, before we regard the distinction as too artificial to include in our annotation scheme?

Structural ambiguities in spoken language

Much of the recent work of my team has dealt with spoken language (we have been compiling the CHRISTINE spoken British English treebank, `http://www.cogs.susx.ac.uk/users/geoffs/RChristine.html`). Indeterminate structural distinctions are particularly noticeable in speech. Rahman & Sampson (2000) drew attention to a number of cases where distinctions that are fundamental with respect to written English turn out to be blurred in the spoken language. For instance, direct v. indirect quotation is conceptually or logically a very clear distinction, which has considerable human significance (relating for instance to different kinds of accuracy obligations on those who quote). In written English the distinction is made very sharp, not just through wording but through punctuation. Yet in spoken English direct v. indirect quotation is not a yes-or-no distinction at all, but at most a cline. The language has several features which mark material as direct quotation or as reported speech, but it is common for these features to be mixed, so that a quotation is more or less direct but not entirely one or the other. A BNC example discussed in Rahman & Sampson (2000) was:

> *well Billy, Billy says well take that and then he'll come back and then he er gone and pay that* KCJ.01053-5

– where, among the underlined items, the introductory *well*, the imperative *take*, and present-tense (*wi*)*ll* rather than (*woul*)*d* point towards direct quotation, but *he* (rather than *I*, referring to Billy) points towards indirect quotation. In spoken English, this kind of direct/indirect quotation ambiguity is so pervasive that it is tempting to see the distinction as an artificial, unrealistic one (so that, in terms of SUSANNE annotation symbols, no contrast would be maintained between `Fn`, for "nominal clause", and `Q`, for "quotation") – though the distinction is so important logically that we did not take this line in the CHRISTINE Corpus.

However, structural ambiguities in speech are not the most significant cases for present purposes. Applying any annotation scheme to spoken language inevitably leads to numerous unclarities caused by the nature of speech rather than the nature of the scheme. Analysts typically work from recordings with little knowledge of the situation in which a conversation occurred or the shared assumptions of the participants. Often, patches of wording are inaudible in the recording; speakers will mis-speak themselves, producing wording which they would not themselves regard as good examples of their language; their "body language" will be invisible to the analyst; and if analysts work from transcriptions, even intonation cues to structure are unavailable. In these circumstances there will often be doubt about how to apply even a very limited, skeleton annotation scheme.

Limits to written-language analytic refinement

The real problem relates to unclarities in applying an annotation scheme to published written language, where the wording is as well disciplined as writer and editor can make it, and the only background assumptions shared by writer and reader are those common to members of their society and hence available to annotators too.

Let me illustrate via a range of examples drawn more or less at random from one written BNC text which I happened to be working with (in connexion with our new LUCY project, `http://www.cogs.susx.ac.uk/users/geoffs/RLucy.html`) at the time of writing this paper. (The sample is extracted from a novel about life in the French Foreign Legion. As English prose, I would judge it to be well-written.)

There are in the first place various passages which are genuinely grammatically ambiguous, e.g.:

> *I had set my sights on getting a good position in training so that I would be sent to the* 2ème Régiment Étranger de Parachutistes. EE5.00933

– is the *so that I …* sequence a constituent of the *sent* clause or the *getting* clause (was being sent to the *Deuxième Régiment* the motive for setting sights, or the potential result of getting a good position)?

> *They were kicked senseless and then handed over to the Military Police who locked them up in the roofless regimental prison before they were handed over to the Colonel of the Regiment for interrogation and questioning.* EE5.00912

– is the *before* clause part of the *locked* relative clause, or is it a constituent of the *then handed over* clause near the beginning (is the handover to the Colonel described as following the prison spell or as following the handover to the Military Police)? In both cases, the alternative interpretations would correspond to different annotation structures in the SUSANNE scheme, and surely in any other plausible linguistic annotation scheme.

Where a passage is genuinely ambiguous, we *expect* an expert to be unable to choose between alternative annotations – that is what "ambiguous" means in this context. Consequently, inability to choose in these cases is not a ground for suspecting that the SUSANNE scheme is over-refined. Notice, though, that even though many linguists would agree that the examples are genuinely ambiguous, these are not the kinds of ambiguity which might be resolved by asking the writer "what he really meant" – in the second case, for instance, the handover to the Colonel in fact followed *both* the handover to the Military Police *and* the prison spell, and there is no reason to suppose that the writer intended one interpretation to the exclusion of the other. This is a frequent situation in real-life usage.

In many other cases, the SUSANNE annotation scheme requires the analyst to choose between alternative notations which seem to correspond to no real linguistic difference (and where the choice is not settled by the rather full definitions of category boundaries that form part of the scheme), so that one might easily conclude that the notation is over-refined – except that the same notational contrast seems clearly desirable for other examples. Here are a handful of instances:

Passive v. BE + *predicative past participle*

The SUSANNE scheme (§4.335) distinguishes the passive construction, as in *I doubt … whether the word <u>can be limited</u> to this meaning …*, from cases where *BE* is followed by a past participle used predicatively, as in … *the powers … <u>were</u> far too <u>limited</u>*. What about the following, in a context where earth is being shovelled over a man:

> *When his entire body was covered apart from*

his head, … EE5.00955

I see no distinction at any level between a passive and a *BE* + predicative particle interpretation of *was covered* here; does that mean that it was a mistake to include the distinction even in connexion with "clear cases" such as those previously quoted?

Phrase headship

The SUSANNE system classifies phrases in a way that depends mainly on the category of their head words, which is commonly uncontroversial. In the example:

> *If we four were representative of our platoon,* … EE5.00859

it is clear that *we four* is a phrase, subject of the clause, but I see no particular reason to choose between describing it as a noun phrase headed by *we*, or a numeral phrase headed by *four*.

Co-ordination reduction v. complete tagma

In the example:

> *He had wound up in Marseilles, sore and desperate, and signed on at Fort St Nicholas*. EE5.00855

the first clause contains a pluperfect verb group *had wound*. It is normal for repeated elements optionally to be deleted from conjoined tagmas, so *signed* might be either the past participle of another pluperfect form from which *had* was deleted, or a past tense forming the whole of a simple past construction. This again seems in this context a distinction without a difference. Yet simple past v. pluperfect, and past tense v. past participle, are elementary English grammatical distinctions likely to be recognized by any plausible annotation scheme.

Interrogative v. non-interrogative how

A subordinate clause beginning with an interrogative is commonly either an indirect question (*I know why* …) or a relative clause (*the place where* …). But if the interrogative form is *how*, there is also a usage in which the clause functions like a nominal clause, with *how* more or less equivalent to *that*:

> … *shouting about the English and how they were always the first to desert* … EE5.00902

> *It was frightening how hunger and lack of sleep could make you behave and think like a real bastard*. EE5.00919

The shouting in the first example was presumably not about the manner of English legionnaires' early desertion but about the fact of it. The second example is more debatable; it might be about either the fact of hunger and no sleep affecting one's psychology, or about the insidious manner in which this occurs. This is an instance where the SUSANNE scheme avoids recognizing a distinction which is arguably real; the scheme does not allow *how* to be other than an interrogative or relative adverb, and therefore treats the *how* clauses as antecedentless relative clauses with *how* functioning as a Manner adjunct, even in the former example. But I could not give a principled reason for failing to recognize a distinction here, when other distinctions that are equally subject to vagueness are required by the annotation scheme.

Multi-word prenominal modifiers

Where a sequence of modifying words precedes a noun head, if the SUSANNE scheme shows no structural grouping then each word is taken as modifying the following word or word-sequence (Sampson 1995: §4.9). But a noun can be premodified by a multi-word tagma, in which case the modifier will be marked as a unit: cf. *He graduated with* [Np [Ns *first class*] *honours* [P *in oil technology*]] … GX6.00022 – *first class* is a noun phrase, the word *first* is obviously not intended as modifying a phrase *class honours*. However, consider the examples:

> *the nearby US Naval base at Subic Bay* EE5.00852
> … *handed over to US Immigration officials* …

EE5.00854

The words *US Naval* could be seen as the adjectival form of *US Navy*, which is a standard proper name; and *US Immigration* is perhaps also current as a way of referring to the respective branch of the American public service. Yet at the same time, the base at Subic Bay is a naval base, and among naval bases it is a US one; and similarly for US immigration officials. If there are no grounds for choosing whether or not to group premodifying words in these cases, does that make it over-refined to recognize such a distinction in cases like *first class honours*?

(In fact the SUSANNE annotation scheme contains an overriding principle that only as much structure should be marked as is necessary to reflect the sense of a passage, and this principle could be invoked to decide against treating *US Naval*, *US Immigration* as units in the examples above. But ideally one would hope that an annotation scheme should give positive reasons for assigning a particular structure and no other to any particular example, rather than leaving the decision to be made in these negative terms.)

It would be easy to give many more examples of structural distinctions which are clear in some cases but seem empty in other cases. Perhaps the examples above are enough to illustrate the point.

I have no definite solution to the problem posed by cases like these. I do not believe that any neat, principled answer is available to the question of how refined a useful general-purpose structural annotation scheme should be; it seems to me that the devising of such schemes will always be something of a "black art", drawing on common-sense rules of thumb and instinct rather than on logical principles.

But, if that is true, it is as well that those of us involved with corpus annotation should be aware that it is so. People who work with computers tend often to be people who expect a logical answer to be available for every problem, if one can find it. For treebank researchers to put effort into trying to establish the "right" set of analytic categories for a language would lead to a lot of frustration and wasted resources, if questions like that have no right answer. The main purpose of the present paper is to urge any who doubt it that, unfortunately, there are no right answers in this area.

Annotation practice and linguistic theory

One group of academics might suggest that there are right answers: namely, theoretical linguists. For theoretical linguists it seems axiomatic that what they are doing in working out the grammatical structure of a language is not devising a useful, workable set of descriptive categories, but *discovering* structure which exists in the language whether linguists are aware of it or not. What makes the structure "correct" is either correspondence to hypothetical psychological mechanisms, or (for linguistic Platonists such as J.J. Katz, e.g. Katz 1981) the fact that languages are seen as mathematical objects with an existence independent of their users.

For some of the problem cases discussed above, it is plausible that linguistic theorizing might yield answers to classification questions which I described as unanswerable. It would not surprise me if some linguistic theory of headship gave a principled reason for choosing one of the words of the phrase *we four* as head. (It would also not be surprising if another linguist's theory gave the opposite answer.) For some other cases it is less easy to envisage how linguistic theory might resolve the issue.

But linguistic annotation ought not to be made dependent on linguistic theorizing, even in areas of structure where theoretical linguists have answers. That would put the cart before the horse. The task of linguistic annotation is to collect and register data which will form the raw materials for theoretical linguistics, as well as for applied natural language processing. If linguistic theory is to be answerable to objective evidence, we cannot wait for the theories to be finalized before deciding what categories to use in our data banks.

The most we can reasonably ask of an annotation scheme is that it should provide a set of categories and guidelines for applying them which annotators can use consistently, so that similar instances are always registered in similar ways; and that the categories should not be blatantly at odds with the theoretical consensus, where there is a consensus. We cannot require that they should be the "correct" categories. To return to the biological analogy: studies of DNA sequences at the end of the 20th century are giving us new information about the theoretically correct shapes of the "family trees" of animal and plant kingdoms. It would have been unfortunate for the development of biology if Linnaeus and his colleagues had waited for this information to become available before compiling their taxonomic system.

A disanalogy with biology

I have alluded to the analogy with biological systematics; questions about how many and what grammatical categories treebankers should recognize have many parallels with questions about how many and what taxa should be recognized by biologists. Since our treebanking enterprise is rather a new thing, it is good to be aware of old-established parallels which may help to show us our way forward.

But although the classification problem is similar in the two disciplines, there is one large difference. We are worse placed than the biologists. For them, the lowest-level and most important classification unit, the species, is a natural class. The superstructure of higher-level taxa in Linnaeus's system was not natural; it was a matter of common-sense and convenience to decide how many higher-order levels (such as genus, phylum, and order) to recognize, and Linnaeus did not pretend that the hierarchy of higher-order groupings corresponded to any reality in Nature – he explicitly stated the contrary (cf. Stafleu 1971: 28, 115ff.). But for most biological purposes, the important thing was to be able to assign individual specimens unambiguously to particular species; the higher-order taxonomy was a practical convenience making this easier to achieve. And species are real things: a species is a group of individuals which interbreed with one another and are reproductively isolated from other individuals. There are complications (see e.g. Ayala 1995: 872-3, who notes that in some circumstances the objective criteria break down and biologists have to make species distinctions by "commonsense"); but to a close approximation the question whether individuals belong to the same or different species is one with a clear, objective answer.

In grammar, we have no level of classification which is as objective as that. So far as I can see, whether one takes gross distinctions such as clause v. phrase, or fine distinctions, say infinitival indirect question v. infinitival relative clause, we always have to depend on unsystematic common sense and *Sprachgefühl* to decide which categories to recognize and where to plot the boundaries between them.

It feels unsatisfying not to have a firmer foundation for our annotation activity. Yet anything which enables us to impose some kind of order and classification on our bodies of raw language data is surely far better than nothing.

References

Ayala, F.J. (1995) "Evolution, the theory of". *Encyclopaedia Britannica*, 15th ed., vol. 18, pp. 855-83.

Katz, J.J. (1981) *Language and Other Abstract Objects*. Rowman & Littlefield (Totowa, New Jersey).

Langendoen, D.T. (1997) Review of Sampson (1995). *Language* 73.600-3.

Meteer, Marie, et al. (1995) *Dysfluency Annotation Stylebook for the Switchboard Corpus*. `http://www.ldc.upenn.edu/myl/DFL-book.pdf`

Rahman, Anna & G.R. Sampson (2000) "Extending grammar annotation standards to spontaneous speech". In J.M. Kirk, ed., *Corpora Galore: Analyses and Techniques in Describing English*, pp. 295-311. Rodopi (Amsterdam).

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

Stafleu, F.A. (1971) *Linnaeus and the Linnaeans*. A. Oosthoek's Uitgeversmaatschappij (Utrecht).

A Statistical Account on Word Order Variation in German

Daniela Kurz
Computational Linguistics, Saarland University
Postfach 15 11 50, 66041 Saarbrücken, Germany
kurz@coli.uni-sb.de

Abstract

In this paper we present a corpus-based study involving the linear order of subject, indirect object and direct object in German. The aim was to examine several hypotheses derived from Hawkins' (1994) performance theory. In this context it was crucial to examine whether and to which extend length influences the order of subject and objects. The analysis was based on data extracted from the annotated NEGRA corpus (Skut et al., 1998) and the untagged Frankfurter Rundschau corpus. We developed an analysis system operating on the untagged corpus that facilitates the acquisition of data and subsequent statistical analysis. In the following, we describe this system and discuss the results drawn from the analysis of the data. These results do not support the theoretical assumptions made by Hawkins. Furthermore, they suggest the investigation of other factors than length.

1 Background and Motivation

Based on the assumption that basic word order regularities are reflected in the frequency of their occurrence, a corpus-based study involving word order phenomena in German was carried out. A number of different parameters have been linked with the linearization of complements and adjuncts in languages exhibiting a relatively free word order. The main factors that have been proposed are: Pronominality, case, information structure, definiteness, thematic roles, stress and length. Theories found in literature range from predominantly competence-based models to explanations almost entirely based on performance assumptions.

Recently, Hawkins' (1994) length-based theory has received much attention in general linguistics (typology), computational linguistics (modelling language evolution) and psycholinguistics (memory-based models of sentence processing). According to Hawkins, the influence of all factors other than length can almost entirely be explained as epiphenomena of length. The data presented by Hawkins are suggestive but much too restricted in size to permit any empirically supported conclusions.

In this paper we will report on a corpus-based study involving six German verbs exhibiting different basic order patterns namely, NOM<DAT, DAT<NOM, ACC<DAT, DAT<ACC . In order to reduce the effects of other factors, we restricted the investigation to non-pronominal NPs in the middle field. Two corpora were chosen for the analysis: a syntactically annotated corpus of German newspaper text, the NEGRA corpus, and the untagged Frankfurter Rundschau corpus. For each of the two verb groups (transitive and ditransitive), pairs of NOM<DAT, DAT<NOM and ACC<DAT, DAT<ACC were considered. The aim was to search for any interdependence of word order and length. Apart from this, we investigated the impact of definiteness on word order. The data acquisition and analysis were to a large part automated.

Section 2 summarizes the main ideas of Hawkins' length-based theory. In section 3 we present the statistical investigation and discuss the results in section 4. Finally we conclude with section 5.

2 Hawkins' Performance Theory

The theory is based on the assumption that limitations on working memory influence the construction of constituents and that humans prefer to arrange constituents in orders that minimize processing effort. Sentence processing is therefore determined by the principle of

Early Immediate Constituents (EIC). Hawkins assumes that phrases are constructed deterministically in a bottom-up fashion. To construct a phrasal node, it is mostly sufficient to recognize a prefix of the new phrase, so that there is no need to wait until all its immediate constituents (ICs) are found. Hawkins postulates that the prefix-based construction of new phrases is triggered by some lexical or phrasal category that uniquely identifies the mother node to be constructed. For example, German and English NPs are recognized at their left periphery, which can be a determiner, an adjective or the head noun. For German (as for English) the main claim of this theory is that all types of phrases that tend to precede their siblings, such as topic phrases, pronominal NPs, complements preceding in basic order and definite NPs are shorter on average than their respective counterparts. This system predicts that example (1) in Figure 2 will be easier to comprehend than example (2), since in the former 4 words have to be parsed instead of 11 in the latter to arrive at the constituent structure of the VP, the NP, and the PP.

In the context of the present study EIC predicts that a short nominative precedes a long dative and that a short dative precedes a long nominative. The same holds for the sequence of accusatives and datives. In addition, Hawkins assumes that verbal position interacts with length in determining the basic word order. In verb-first and verb-second sentences "short before long" should be strongly preferred, in verb-final sentences "short before long" should be slightly preferred. Hence, the following parameters were investigated in this study:

- length of the NP;
- verbal position;
- definiteness.

Due to lack of space, we will focus in what follows on parameter length and its interaction with definiteness. Kurz (2000) presents the investigations of all parameters in detail.

3 Statistical Investigation

Statistical analysis was performed in two steps:

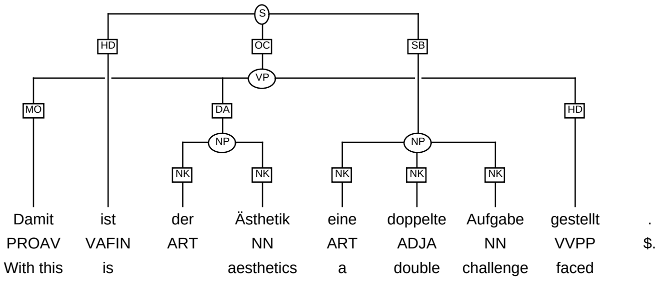
- Determining the verbs to be investigated by extracting all sentences that exhibit the relevant word pattern (NOM<DAT, DAT<NOM, ACC<DAT, DAT<ACC) from the NEGRA corpus.
- Determining the frequencies of the word patterns each verb occurs with using the much larger Frankfurter Rundschau corpus. The distribution found in this corpus was then statistically analysed.

3.1 Extraction of Word Patterns

In order to determine a set of interesting verbs by pursuing an empirical approach, the NEGRA corpus was chosen. The NEGRA corpus is a treebank currently consisting of 20 000 sentences or 355 000 tokens. The annotation scheme of the treebank combines phrase-structures and dependency-based schemes (cf. (Skut et al., 1997). Three types of information are encoded:

- predicate argument structure: trees with possibly crossing branches;
- syntactic categories: node labels and part-of-speech tags;
- functional categories: edge labels.

The representation format is shown in the figure[1] below:



With this aesthetics is faced with a double challenge

Figure 1: Encoding of a sample structure

All relevant word order patterns have been extracted by implementing matching routines operating on the structure of the corpus. The determination of the middle field was achieved by a decision tree dealing with the possible patterns displayed in Table 1.

[1] Edge labels: HD head, OC clausal object, SB subject, MO modifier, DA dative, NK noun kernel. Crossing edges indicate discontinuous constituency.

(1)	I	$_{\text{VP}}$[gave	$_{\text{PP}}$[to	Mary]	$_{\text{NP}}$[the	valuable	book	that	was	extremely	difficult	to	find]]
		1	2	3	4								

(2)	I	$_{\text{VP}}$[gave	$_{\text{NP}}$[the	valuable	book	that	was	extremely	difficult	to	find]	$_{\text{PP}}$ [to	Mary]]
		1	2	3	4	5	6	7	8	9	10	11	

Figure 2: Recognition of phrasal categories

left sentence bracket	**right sentence bracket**
finite verb	empty or separable verbal prefix
auxiliary or modal	verbal complex e.g. perfect participle
complementizer	finite verb or verbal complex

Table 1: Possibilities of filling the sentence brackets

The positions of the left and the right sentence bracket restrict the scope in which possible NP sequences can occur. In Figure 1 the left sentence bracket consists of the finite auxiliary *ist* and the right sentence bracket is represented by the perfect participle *gestellt*. The middle field thus consists of the dative NP *der Ästhetik* and the nominative NP *eine doppelte Aufgabe*. Verbs for further investigation have been chosen according to their frequency of occurrence in each word order variation. The verbs which occurred most frequently have been selected. Tables 2 and 3 show the selected verbs and their distribution in the NEGRA corpus.

Verb	**Total**	**DAT<NOM**	**NOM<DAT**
gelingen (to succeed)	43	4	0
helfen (to help)	70	0	5
zur Verfügung stehen (to be available)	31	3	2

Table 2: Distribution of transitive verbs in the NEGRA corpus

Verb	**Total**	**DAT<ACC**	**ACC<DAT**
geben (to give)	560	16	0
vorstellen (to present)	38	0	3
zur Verfügung stellen (to make available)	24	0	3

Table 3: Distribution of ditransitive verbs in the NEGRA corpus

The second column (headed by Total) shows the numbers of all sentences containing the respective verb. In columns headed by DAT<NOM, NOM<DAT, ACC<DAT and DAT<ACC the numbers of sentences exhibiting each of the relevant patterns are given. It is evident that only a small part of the items found meets the search conditions. This is because the search was restricted to full NPs and for the most part one of the relevant NPs was pronominal.

Because of the low frequency of any individual verb in the NEGRA corpus we analysed the much larger untagged Frankfurter Rundschau corpus for the verbs under investigation.

3.2 Mining the Untagged Frankfurter Rundschau Corpus

The untagged Frankfurter Rundschau corpus consists of raw ASCII data. It contains 1.644 million sentences or 40.9 million tokens. Considering the size of the data it was crucial to automate the analysis as far as possible. This was achieved by developing an evaluation system making use of existing NLP tools and redefined interfaces. With this system, it was possible to ascertain, handle, and analyse the data with a minimum of manual revision.

The executed steps are shown in Figure 3.

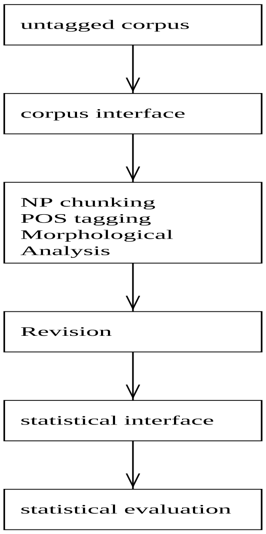


Figure 3: Flow chart of executed steps

3.2.1 Corpus Interface

The extraction of all sentences containing one of the verbs under investigation was done by running pattern-matching routines. These routines used regular expressions including all verbal inflections and carried out format conversions required by the subsequent component.

3.2.2 Shallow Parsing and Morphological Analysis

As already mentioned we were looking for particular sequences of case-marked NPs. Therefore each NP had to be labeled with case information. In order to do this, the NP boundaries had to be determined. We employed a stochastic parser (Skut's (1999) chunk tagger) that recognizes the internal structure of phrases and determines NP boundaries. As output for complex NPs, the chunk tagger delivered information about phrase boundaries and part-of-speech tags. For the annotation of structural and part-of-speech information, the chunk-tagger uses two instances of the TnT-Tagger developed by Brants (1996). The next step was followed by a morphological analysis (MORPHIX (Finkler and Neumann, 1986)) labeling each word of each NP with information about inflection based on the already available part-of-speech information. The case determination of complex NPs was done by a unification of the number, gender and case values of each word belonging to the whole NP. The output consisted of NPs labeled with partly ambiguous case information as shown in examples (3)-(5).

(3)	[NOM ACC	Die	Gemeinde]
		The	community

(4)	[NOM GEN DAT ACC	Kinder]
		children

(5)	[DAT	einem	Nachrichtenmagazin]
		a	news magazine

3.2.3 Revision

In the following revision, all sentences containing at least one of the relevant NPs in the middle field were automatically extracted. The case disambiguation, the determination of length and definiteness of each NP, and the determination of the verbal position was done manually. An additional program interface recoded the results of this revision in numeric variables required by the statistical interpretation.

3.2.4 Evaluation

For the evaluation of the system we compared its output with a manually evaluated sample containing 100 sentences and determined precision and recall. Table 4 shows precision and recall for the chunking application and the morphological analysis.

	Precision	Recall
Chunk tagger	92%	84%
MORPHIX	53%	100%

Table 4: Recall and precision for used NLP tools

The recall of 100% for MORPHIX has several reasons. Firstly MORPHIX is operating only on the correct output of the preceding chunk tagger; secondly, in each case the unification fails MORPHIX labels the NP with [NOM, DAT, ACC, GEN]. This perfect recall is responsible for the low precision of 53% because of the frequently occurring fourfold case assignment. However, this behaviour of MORPHIX exactly met our requirements, since we were primarily concerned with extensive data acquisition (the automatic analysis was followed by a manual evaluation anyway.)

4 Results

The analysis of the data was done by using cross tables, chi square tests and the analysis of means. In spite of the corpus size, the data do not permit chi square tests in some cases. In these cases, we pursued a purely descriptive approach. For the sake of space, we will describe the results of the cross tables only. We will first have a look at the distribution of NP sequences in the Frankfurter Rundschau corpus. In addition we will consider the factors length and definiteness in isolation. We will then look at the interaction of the factors holding definiteness constant.

4.1 Distribution in the Frankfurter Rundschau Corpus

Table 7 shows the distribution of the examined sequences for each transitive verb, Table 8 shows the frequencies of the object sequences for ditransitive verbs.

By looking at the distribution of the NP orderings there are concentrations on either one

Verb	Basic Order	EIC-un-marked	EIC-marked	Rearrangement	EIC-un-marked	EIC-marked	Total EIC-un-marked	Total EIC-marked
gelingen	257	58%	21%	3	100%	0%	58%	20%
helfen	148	45%	30%	13	39%	23%	44%	29%
zur Verfügung stehen	119	63%	18%	37	41%	24%	54%	24%

Table 5: Transitive verbs and EIC

Verb	Basis Order	EIC-un-marked	EIC-marked	Rearrangement	EIC-un-marked	EIC-marked	Total EIC-un-marked	Total EIC-marked
geben	457	40%	37%	8	38%	38%	40%	37%
vorstellen	29	69%	14%	24	8%	59%	42%	34%
zur Verfügung stellen	182	47%	29%	78	27%	42%	41%	33%

Table 6: Ditransitive verbs and EIC

Verb	verb frequency	DAT<NOM	NOM<DAT
gelingen	3980	**257**	3
helfen	5700	13	**148**
zur Verfügung stehen	1974	**119**	37

Table 7: Transitive verbs

Verb	verb frequency	DAT<ACC	ACC<DAT
geben	11354	**457**	8
vorstellen	3694	29	24
zur Verfügung stellen	2094	**182**	78

Table 8: Ditransitive verbs

of the two possible sequences (marked boldface) for each verb. *Gelingen* and *zur Verfügung stehen* clearly favour DAT<NOM ordering while *helfen* shows the opposite preference. A similar picture is found for the ditransitive verbs, with one exception: *vorstellen*, both sequences are represented nearly the same. *Geben* and *zur Verfügung stellen* show a clear preference for DAT<ACC ordering. Given this distribution, it seems more appropriate to determine basic word order dependent on the particular verb, rather than specifying a *general* basic order of arguments (cf. (Haider, 1993)). Thus, we consider orderings exhibiting high frequencies as basic orders and orderings exhibiting low frequencies as rearrangements, e.g. the basic order of *gelingen* is DAT<NOM, the rearrangement is NOM<DAT, *helfen* appears with NOM<DAT as basic order and DAT<NOM as rearrangement. For *vorstellen*, the issue which ordering is the basic order and which ordering is the rearrangement cannot be determined from the empirical distribution, since this verb is roughly equi-based with respect to NP-order.

Against the background of Hawkins' model the question arises if both basic order and rearrangement can be motivated by length phenomena.

4.2 How Good Are EIC's Predictions?

Tables 5 and 6 show the total percentages of the EIC-unmarked and EIC-marked cases, the basic order and the rearrangement of each verb. The EIC-unmarked cases indicate those for which EIC makes the right predictions (short NP precedes long NP), in the EIC-marked cases EIC makes the wrong predictions (long NP precedes short NP). The proportions of the marked and unmarked cases do not add up to 100% because the cases with two NPs of equal length have not been considered.

In total we can observe that the EIC predictions are fairly good for the unmarked cases but still there is a considerable amount of cases deviating from EIC. Apart from this the EIC predictions for rearrangements are worse than for the basic order. This becomes clear from the pattern of *zur Verfügung stehen* of Table 5 and *zur Verfügung stellen* of Table 6. Comparing the EIC-unmarked and the EIC-marked cases of the basic orders with the EIC-unmarked and EIC-marked cases of the rearrangements, the proportions of the unmarked cases are lower and the proportions of the marked cases are higher in the rearrangements. The same holds for *vorstellen.* We may not determine basic order and rearrangement but the EIC predictions are quite bad for one of the two orderings (shown in the column headed by rearrangement).

Since the rearrangements of *helfen, gelingen* and *geben* are underrepresented (*gelingen*: 3, *helfen*: 13 and *geben*: 8), there is no evidence in favour or against the described observation.

Verb	Basic Order				Rearrangement			
	def-indef	indef-def	def-def	indef-ndef	def-indef	indef-def	def-def	indef-ndef
gelingen	36%	2%	62%	-	-	-	100%	-
helfen	16%	19%	62%	3%	39%	-	55%	8%
zur Verfügung stehen	72%	-	16%	12%	16%	5%	79%	-

Table 9: Transitive verbs and definiteness

Verb	Basic Order				Rearrangement			
	def-indef	indef-def	def-def	indef-indef	def-indef	indef-def	def-def	indef-indef
geben	56%	7%	26%	11%	38%	12%	50%	-
vorstellen	17%	3%	73%	7%	8%	-	92%	-
zur Verfügung stellen	66%	1%	18%	15%	7%	19%	69%	5%

Table 10: Ditransitive verbs and definiteness

To conclude, length phenomena do not seem to be the only reason for deviation from the basic order. The results suggest to concentrate on additional factors.

4.3 Definiteness

Tables 9 and 10 show the distribution of the sequences of definite and indefinite NPs in our data set.

For the basic order of both verb groups it can be observed that def-indef (definite NP precedes indefinite NP) and def-def sequences (definite NP precedes definite NP) occur most frequently. For the rearrangements, it is striking that most of the sequences belong to the def-def pattern. This holds for the transitive as well as for the ditransitive verbs. One exception can be found: the indef-def sequences of *zur Verfügung stellen*. Since Hawkins claims that all factors apart from length are epiphenomenal, the effect of length should be even stronger if the other factors (e.g. definiteness) are held constant. In our analysis, this means that we should see a clear effect of EIC for the rearranged groups because of the high amount of def-def sequences. This, on the other hand, is at odds with the results we have already drawn from Tables 5 and 6. Recall that, EIC made worse predictions for the rearrangements than for the basic order.

4.4 EIC Revisited

To test the expectations mentioned above we evaluated the EIC predictions for rearranged def-def sequences. The results are listed in Tables 11 and 12.

Verb	Rearrangement	def-def	EIC unmarked	EIC marked	Total EIC unmarked	Total EIC marked
gelingen	NOM<DAT	3	100%	-	58%	20%
helfen	DAT<NOM	7	57%	14%	44%	29%
zur Verfügung stehen	NOM<DAT	29	21%	43%	54%	24%

Table 11: Transitive verbs: EIC and definiteness

Verb	Rearrangement	def-def	EIC unmarked	EIC marked	Total EIC unmarked	Total EIC marked
geben	ACC<DAT	4	50%	25%	40%	37%
vorstellen	ACC<DAT	22	9%	55%	42%	34%
	DAT<ACC	21	71%	10%	42%	34%
zur Verfügung stellen	ACC<DAT	54	30%	41%	41%	33%

Table 12: Ditransitive verbs: EIC and definiteness

The contribution of EIC does not meet the expectation derived from the "epiphenomenon hypothesis". Furthermore, the data demonstrate that EIC makes incorrect predictions for verbs exhibiting a large proportion of rearranged def-def sequences (*zur Verfügung stehen, vorstellen, zur Verfügung stellen*). For the verbs with few rearranged def-def orderings (*gelingen, helfen, geben*), EIC makes fairly good predictions. Comparing the definite, rearranged EIC-marked cases with the total percentages of the EIC marked cases (serving as baseline) the proportions of the definite, rearranged EIC marked cases are above baseline for the verbs exhibiting high frequencies. Comparing the definite, rearranged EIC-unmarked cases with the total percentages of the EIC-unmarked cases the proportions of the definite, rearranged EIC-unmarked cases are below baseline. Again, this supports the assumption that EIC does not dominate rearrangements. The results indicate that a closer examination of information-based parameters such as *Topic* and *Focus*, which are correlated with definiteness, will be required.

5 Conclusions

The results presented in this paper suggest that:

- The determination of basic order and rearrangement depends on the particular verb. The data do not support specification of a *general* basic order.
- EIC is not the primary factor determining the linearization of complements in the middle field.
- Considering rearranged sequences the factor definiteness dominates EIC.

The made observations are at odds with Hawkins' (1994) claiming that length is the only factor determining word order. For example definiteness determines word order even in those cases where EIC cannot motivate the ordering. Moreover, several parameters seem to interact and determine the sequence to a different extent, a claim that has already been proposed by Uszkoreit (1987).

The present results emphasize the necessity of further empirical research based on interpreted and uninterpreted corpora. Especially an examination of the interaction between definiteness and information-based factors requires further extensive corpus-based studies.

The insights gained from these methodologies show that linguists cannot rely exclusively on introspective judgements as their sole source of data. Furthermore, we hope to have demonstrated the productivity of employing corpus based studies in syntactic research.

6 Acknowledgements

This research was funded by the DFG Sonderforschungsbereich 378 Resource-Adaptive Cognitive Processes, Project C3 Concurrent Grammar Processing. I am grateful to Markus Becker, Stephan Oepen, Oliver Plaehn, Christoph Scheepers, Hans Uszkoreit and Feiyu Xu for constructive comments and their contributions.

References

Thorsten Brants. 1996. TnT – A Statistical Part-of-speech Tagger. Technical report, Saarland University.

W. Finkler and G. Neumann. 1986. Morphix – ein hochportabler Lemmatisierungsmodul für das Deutsche. Technical report, Saarland University, FB Informatik.

Hubert Haider. 1993. *Deutsche Syntax - generativ.* Gunter Narr Verlag, Tübingen.

John A. Hawkins. 1994. *A Performance Theory of Order and Constituency.* CUP, Cambridge.

Daniela Kurz. 2000. Wortstellungspräferenzen im Deutschen. Master's thesis, Saarland University.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An Annotation Scheme for Free Word Order Languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing ANLP-97*, pages 88–95, Washington, DC.

Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A Linguistically Interpreted Corpus of German Newspaper Text. In *Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation*, pages 18–24, Saarbrücken, Germany.

Wojciech Skut. 1999. *Partial Parsing for Corpus Annotation and Text Processing.* Ph.D. thesis, Saarland University.

Hans Uszkoreit. 1987. *Word Order and Constituent Structure in German*, volume 8 of *Lecture Notes.* CSLI, Stanford.

Bottom–Up Tagset Design from Maximally Reduced Tagset

Péter Dienes and **Csaba Oravecz**
Research Institute for Linguistics
Hungarian Academy of Sciences, Budapest
{dienes,oravecz}@nytud.hu

Abstract

For highly inflectional languages, where the number of morpho-syntactic descriptions (MSD) is very high, the use of a reduced tagset is crucial for reasons of implementation problems as well as the problem of sparse data. The standard procedure is to start from the large set of MSDs incorporating all morphosyntactic features and design a reduced tagset by eliminating the attributes which play no role in disambiguation. This paper presents the opposite approach which using a greedy algorithm maximally reduces a tagset without loss of information, and instead of elimination, re-introduces features. This process can arrive at a very small tagset and result in accuracy comparable to that achieved with larger tagsets designed by elimination. The language model based on the reduced tagset needs fewer parameters and training time decreases significantly.

1 Introduction

In highly inflectional languages, the number of morpho-syntactic descriptions (MSD), required to descriptionally cover the content of a word-form lexicon, tends to rise quite rapidly, approaching a thousand or even more set of distinct codes. For the purpose of automatic disambiguation of arbitrary written texts, using such large tagsets would raise very many problems, starting from implementation issues of a tagger to work with such a large tagset to the more theory-based difficulty of sparseness of training data. Tiered tagging (Tufiş, 1998) is one way to alleviate this problem by reformulating it in the following way: starting from a large set of MSDs, design a reduced tagset, Ctag-set, manageable for the current tagging technology. The standard procedure is to start from the large set of MSDs incorporating all morphosyntactic features and design a reduced tagset by eliminating the attributes which play no role in disambiguation. However, there are a number of reasons for which one can question whether such a process can produce anything close to an optimal tagset and eliminate all irrelevant features. In section 2 we briefly outline these reasons and in section 3 present the data used. Section 4 suggest and alternative approach that just takes the opposite way and a maximally reduced tagset as starting point for the design process. Section 4.2 will present some preliminary results on tagging accuracy and error analysis comparing the performance of the tagging process with tagsets of different cardinality. Conclusions and suggestions for further work will follow in section 5.

2 Tagset design and highly inflected languages

The combinatorial possibilities of inflection and derivation in highly inflectional languages pose a challenge for corpus annotation in that it is difficult to establish a set of morphosyntactic descriptions that does justice to the rich morphosyntactic information encoded within the words and at the same time remains computationally tractable. The design process of a reduced tagset has to consider two fundamental requirements: to identify and leave out the features/values in the MSDs which do not provide relevant clues for the contextual disambiguation, and to make it possible to recover as accurately and fast as possible the information eliminated in the previous phase.

The standard approach is usually a trial-and-error one augmented by some algorithm and relies both on human introspection and evidence provided by the data analysis (Elworthy, 1995), (Chanod and Tapanainen, 1995), (Tufiş, 2000).

One can use an information loss-less algorithm to convert the MSD-set into a Ctag-set which might reduce the size of the tagset with 10-20% (Brants, 1995); however, this is too little for a large initial tagset. Modifying such an algorithm to allow for limited ambiguity (that is losing a limited amount of information), could result in a drastic reduction of the Ctag-set, up to a cardinality which is within the restrictions imposed by the available training data and computing power (Tufiş, 1998). Nevertheless, this procedure fails to obtain the optimum result for at least two reasons: there is, even if limited, loss of information and the recoverability of information contained in the original MSDs is not preserved; and features that do not appear in ambiguity classes are usually not submitted to the reduction algorithms and may be preserved unnecessarily.

3 Data analysis

The language resource of our analysis consisted of the whole current stock of the Hungarian National Corpus (approximating 80m words) compiled into a word frequency list as input to the morphological analysis. The initial assumption is that this large number of word forms contain all possible ambiguity classes that can occur in the language. Table 1 presents some basic statistics on the range of word form variation found in the corpus.

Entries	Word forms	Lemmas
74,063,211	1,728,771	429,612[1]

Table 1: The distribution of word forms

The word form list was processed with the morphological analyzer developed originally for Hungarian (Prószéky and Tihanyi, 1996). An MSD notation was constructed which represented the POS category and the inflectional structure of the word, and which can in principle be mapped into the EAGLES compliant encoding scheme developed in Multext–East (Erjavec and Monachini, 1997). The MSD scheme, as an initial step in tagset creation, was converted into an attribute/value single string representation. The intent at this stage was merely to preserve in a concise and consistent notation all the information provided by the MSD that is relevant for tagging. Table 2 displays the features encoded in this initial Ctag scheme (Full set) for the major POS categories.

As the cardinality of the full initial tagset was too high to be handled by current tagging methods (2148), especially by statistical taggers, a medium tagset was designed by feature elimination as detailed in (Tufiş et al., 2000). (This medium tagset does not ensure full recoverabilty, though.) These two tagsets serve as the basis of comparison in the evaluation of the alternative approach for tagset creation we will propose below. In the experiments, two HMM taggers are used: Thorsten Brants' trigram TnT tagger (Brants, 1998) and the MULTEXT-ISSCO (M–I) bigram tagger (Gilbert and Amstrong, 1995) used in the Multext-East project (Erjavec and Ide, 1998). The training corpus consists of two register-diverse corpora: the first three quarters of Orwell's 1984 and newspaper text, adding up to 87969 tokens altogether. The test corpus includes the rest of the Orwell and newspaper texts, 21267 tokens in total. The MULTEXT-ISSCO tagger is trained with the Baum-Welch algorithm. The TnT tagger has the problem of learning possible ambiguity classes and words from the training corpus only. To remedy this situation, after the training phase, we enriched the generated lexicon file with further ambiguities and added words from the test corpus with their ambiguity classes. The tagging results with the above tagsets are presented in Tables 3 and 4.

	Error perc.	Error rate	Perf.
M-I	23.83%	6.04%	93.96%
TnT	14.00 %	3.55%	96.45%

Table 3: Full tagset (2148 tags)

4 Maximal reduction and bottom–up design

4.1 Maximal reduction of the tagset

The alternative approach using a greedy algorithm maximally reduces a tagset without loss

[1] The number of lemmas were calculated on the assumption that alternatives in ambiguous cases were evenly distributed. This is obviously false but the correct figure could only be arrived at after the corpus has been completely disambiguated.

POS	Num	Pers	Stem [NAR] Mood/Tense [V]	Case [N] Def [V]	Owner's Num	Owner's Pers	Total
N	2 [PS]	3 [123]	5 [QAVNP]	21	2 [PS]	3 [123]	2058*
A			2 [AV]				2*
R			2 [RV]				2*
V	2 [PS]	3 [123]	5 [PRCSI]	3 [ID2]			79*
Invariant minor categories: Q, D, PRE, RP, C, Int, Y							7
							2148

N = Noun	A = Adjective	R = Adverb	V = Verb
Q = Numeral	D = Article	PRE = Verbal prefix	RP = Postposition
C = conjunction	Y = Abbreviation	Int = Interjection	

Def = Agreement in definiteness with object (def, indef, 2nd person)
Owner's Num = sing. or plural owner   Owner's Pers = person marker of owner
* = not all combinations are possible, so not a simple product
[NAR][V][N] = POS categories to which the attribute apply

Table 2: The initial Ctag scheme (F set)

	Error perc.	Error rate	Perf.
M-I	22.48%	5.70%	94.3%
TnT	13.37%	3.39%	96.61%

Table 4: Medium tagset (240 tags)

of information, and instead of elimination, reintroduces features. This process first arrives at a very small tagset and the application of this tagset in tagging results in a dramatic drop in accuracy compared to that achieved with a tagset designed from MSD reduction with the elimination algorithm in (Tufiş, 2000) and linguistic introspection. However, even the reintroduction of a few morphosyntactic features leads to a sharp increase in accuracy comparable to that achieved with larger tagsets designed by elimination. The language model based on the reduced tagset needs fewer parameters and training time decreases significantly.

The construction of the minimal tagset proceeds the following way. First a graph G is established whose vertices are the tags of the initial tagset. Two points (tags) are connected with an edge if and only if there exists a word which can be assigned both tags. That is, two tags are not connected if they do not occur in an ambiguity class. Then, a partition of this graph is created as follows:

> x and y are in the same partition if and only if there is no (x, y) edge.

The problem is equivalent to the colourability problem of the graph G:

Colourability problem: The aim is to colour the vertices of a graph G with as few colours as possible so that neighbouring vertices have different colours.

In the general case the problem of finding the minimal number of colours (*chromatic number*, $\chi(G)$) cannot be solved within polynomial time. Nevertheless, certain estimations of $\chi(G)$ can be given. The algorithm to be discussed and applied here, for example, yields the result:

$$\chi(G) \leq 1 + \max_{g \in G} \phi(g),$$

where $\phi(g)$ is the degree of the point g, i.e. the number of its neighbours. The algorithm is a simple greedy algorithm. The colours are nonnegative integers.

Algorithm:

1. *Ordering phase*: order the vertices of the graph in any way;
2. *Colouring phase*: for each $i = 1, 2, \ldots$ colour the ith vertex with the smallest available colour. Make this colour unavailable for all neighbouring vertices.

In fact, according to *Brooks-theorem* the *chromatic number* can easily be decreased by one, i.e. (Gross and Yellen, 1998):

Brooks-theorem.

$$\chi(G) \leq \max_{g \in G} \phi(g)$$

Now, consider the graph obtained from the 74-million-word wordlist, tagged with the full tagset. Out of the 1105 tags 968 occur in ambiguity classes, the maximal degree of the vertices of the graph is 192. According to the above theorems, this means that the tagset can be reduced to 192 tags without merging ambiguity classes. This in itself is quite a considerable decrease in the number of tags.

However, for graphs containing several vertices, the estimations obtained from these theorems might lie far over the actual value of the *chromatic number*. This might especially be the case if we deal with graphs obtained from natural language corpora, because these graphs seem to be unsaturated. Figure 1 presents the top 20 degrees of the "Hungarian graph".

1	NS3NN	192
2	R	184
3	VS3RI	71
4	P	57
5	NS3NA	54
6	AS_A	54
7	RP	49
8	NP3NN	47
9	NS3NP	39
10	NS3NS	36
11	NS3PC	36
12	NS3NNS3	36
13	AS_V	35
14	NS3ND	32
15	NS3NI	31
16	NS3N2	31
17	Z	29
18	NS3NX	29
19	NS3NT	28
20	NS3N3	28

Figure 1: Degree of vertices

The data clearly shows that there are two vertices with fairly large number of points, but the degree of vertices decreases rapidly. This might suggest that this graph can be coloured with relatively few colours. Indeed, the actual experiment with the algorithm described above yielded a surprising result: the graph can be coloured with 10 colours, that is, the number of tags can be reduced to 10 without merging ambiguity classes and retaining full recoverability.

4.2 Enriching the minimal tagset

The minimal tagset containing only 10 tags significantly reduces the problem of sparse data. However, with the radical reduction of the tagset, though recoverability is retained, we have lost important environmental information which could serve as tagging clues for the tagger. Thus, as illustrated in Tables 3 and 5, we face radical decrease in tagging accuracy even with respect to the results exhibited by the full tagset. (The cardinality of the tagset is indicated in parentheses.)

	Error perc.	Error rate	Perf.
M-I	32.78%	8.31%	91.69%
TnT	60.94%	15.45%	85.55 %

Table 5: Minimal tagset (10)

The inaccuracy originating from the minimal tagset is especially spectacular in the case of the HMM-based trigram TnT tagger. Here, 15.54% of all words is mistagged, which is over 60% error on ambiguous words. The decrease in the actual performance of the MULTEXT-ISSCO tagger is less conspicuous, though still significant.

One important problem with the minimal tagset is that it fails to indicate punctuation, that is, punctuation tags (CPUNCT, OPUNCT, SPUNCT and WPUNCT) are merged with each other and several other tags. The increase in the performance of the TnT tagger is significant if these four tags are retained. This is illustrated in Table 6.

	Error perc.	Error rate	Perf.
M-I	31.81%	8.06%	91.94%
TnT	18.81%	4.77%	95.23%

Table 6: Minimal tagset with punctuation tags (14)

Interestingly, the reactions of the MULTEXT-

ISSCO tagger to this small change is less radical: the bigram HMM-base tagger seems to depend less on the information provided by punctuation tags. One possible reason for the difference of the behavior between the two models can be that information before the punctuation mark is unavailable for the bigram tagger, regardless whether it "knows" that the word to be disambiguated is preceded by a punctuation mark. On the other hand, a trigram tagger can "learn" to disregard punctuation tags and consider the previous tags only. Whether this assumption can emprically be justified, however, is subject to careful future research.

Another clue that can help the proper identification of tags is the distribution of the main categories, i.e. nouns, verbs and adjectives. This type of information is especially useful for the bigram tagger, for reasons discussed above (cf. Table 7).

	Error perc.	Error rate	Perf.
M-I	19.66%	4.98%	95.02%
TnT	14.84%	3.76%	96.24%

Table 7: Minimal tagset NAV heads only (30)

As we can see, this information provided by the re-introduction of the main head categories proves to be crucial for the trigram tagger as well. Note that the performance of the MULTEXT-ISSCO tagger with these 30 tags is higher than the performance with the handcrafted, "linguistically motivated" medium tagset.

The combination of the two types of information does not increase the performance of the tagger significantly. Similarly, with the re-introduction of all head categories, the error of the taggers does not decrease crucially, as is illustrated in Table 8.

	Error perc.	Error rate	Perf.
M-I	18.94%	4.80%	95.2%
TnT	14.35%	3.64%	96.36%

Table 8: Minimal tagset with all head categories (39)

Hungarian has a very rich case system with 22 cases, which might offer important tagging clues in the disambiguation process. In the experiment, in order to avoid the proliferation of tags, we reduced the possible morphological cases to three: nominative, accusative and other case. The results thus obtained are of considerable importance: though the performance of the bigram taggers decreases insignificantly, the trigram tagger's performance reaches the performance shown with the hand-crafted medium tagset.

	Error perc.	Error rate	Perf.
M-I	19.12%	4.85%	95.15%
TnT	13.54%	3.43%	96.57%

Table 9: Minimal tagset with head cat.s and N case (59)

However, these results are only preliminary inasmuch as only considerably larger training and test corpora and much more extensive testing could provide reliable justification for the re-introduction of one or the other features. Still, these preliminary experiments indicate that a bottom-up procedure can perform at a similar level to a top-down eliminative approach.

5 Conclusion

The paper described a method of maximally reducing a tagset which is supplemented by a "bottom–up" procedure of re-introduction of features, which can achieve acceptable tagging accuracy using a very small tagset with full MSD recoverability. This method is based on a fast and effective algorithm and not only leads to building a language model with fewer parameters in a comparably shorter training time but could also give insight to finding those morphosyntactic features that provide relevant information as contextual clues in ambiguity resolution.

Further investigation should involve more types of taggers including a rule based application (Alexin et al., 1999) as well. It would also be interesting to see how far tagging performance can be improved by this method[2], and extend the experiments to other languages where the MSD cardinality and the size of the

[2] Present tagger implementations cannot produce above around 96% for Hungarian, which constitutes an actual limit for testing this method.

tagset used in tagging experiments is high (Harris et al., 2000), (Hajič and Hladka, 1998). Another crucial advantage lies in the possibility of algorithmic feature re-introduction, the problem of which should also be addressed in the future.

References

Zoltán Alexin, Tamás Váradi, Csaba Oravecz, Gábor Prószéky, János Csirik, and Tibor Gyimóthy. 1999. FGT – a framework for generating rule-based taggers. In *ILP-99 Late-Breaking papers*, Bled, Slovenia.

Thorsten Brants. 1995. Tagset reduction without information loss. In *Proceedings of ACL–95*, Cambridge, MA.

Thorsten Brants, 1998. *TnT – A Statistical Part-of-Speech Tagger, Instalation and User Guide*. University of Saarland.

Jean-Pierre Chanod and Pasi Tapanainen. 1995. Creating a tagset, lexicon and guesser for a french tagger. In E. Tzoukermann and S. Armstrong, editors, *From Texts to Tags: Issues in Multilingual Language Analysis: Proceedings of the ACL SIGDAT Workshop*, pages 58–64, Geneva.

David Elworthy. 1995. Tagset Design and Inflected Languages. In *Proceedings of the ACL-SIGDAT Workshop*, Dublin. (also available as cmp-lg/9504002).

Tomaž Erjavec and Nancy Ide. 1998. The MULTEXT-EAST corpus. In Antonio Rubio, Natividad Gallardo, Rosa Castro, and Antonio Tejada, editors, *First International Conference on Language Resources and Evaluation, LREC'98*, pages 971–974, Granada. ELRA.

Tomaz Erjavec and M. Monachini. 1997. Specifications and notation for lexicon encoding. COP Project 106 Multext-East, Deliverable D1.1 F (Final Report).

R. Gilbert and S. Amstrong. 1995. Tagging tool. MULTEXT Deliverable 2.4.1.

Jonathan Gross and Jay Yellen. 1998. *Graph Theory and Its Applications*. CRC Press.

Jan Hajič and Barbora Hladka. 1998. Tagging inflective languages: prediction of morphological categories for a rich structured tagset. In *Proceedings of the 36_{th} annual meeting of the ACL – COLING*, Montreal, Canada.

Papageorgiou Harris, Prokopidis Prokopis, Giouli Voula, and Piperidis Stelios. 2000. A unified PoS tagging architecture and its application to Greek. In *Proceedings of Second International Conference on Language Resources and Evaluation*, Athens.

Gábor Prószéky and László Tihanyi. 1996. Humor – a Morphological System for Corpus Analysis. In *Proceedings of the first TELRI Seminar in Tihany*, pages 149–158, Budapest.

Dan Tufiş, Péter Dienes, Csaba Oravecz and Tamás Váradi. 2000. Principled hidden tagset design for tiered tagging of Hungarian. In *Proceedings of Second International Conference on Language Resources and Evaluation*, Athens.

Dan Tufiş. 1998. Tiered tagging. Technical Report 32, RACAI.

Dan Tufiş. 2000. Using a large set of eagles-compliant morpho-syntactic descriptors as a tagset for probabilistic tagging. In *Proceedings of Second International Conference on Language Resources and Evaluation*, Athens.

The Detection of Inconsistency in Manually Tagged Text

Hans van Halteren
Dept. of Language and Speech
University of Nijmegen
P.O. Box 9103
6500 HD Nijmegen
The Netherlands
hvh@let.kun.nl

Abstract

This paper proposes a method to detect the presence of inconsistency in a given manually tagged corpus. The method consist of generating an automatic tagger on the basis of the corpus and then comparing the tagger's output with the original tagging. It is tested using the written texts from the BNC sampler and a WPDV-based tagger generator, and shown to be both an efficient method to derive a qualitative evaluation of consistency and a useful first step towards correction.

1 Introduction

Wordclass tagged corpora are a very popular resource for both language engineers and linguists. If these corpora are used for inspiration and exemplification, size may be more important than quality and a fully automatically tagged corpus can suffice. For other uses, quality is of much higher importance, and here there will generally be a preference for manually corrected corpora, even though they may be smaller. However, manual correction means human involvement, and that again means a much higher potential for inconsistency (cf. e.g. Marcus et al. (1993); Baker (1997)).

Before we go and base our NLP systems or linguistic theories on the wordclass tags found in a tagged corpus, then, it would certainly be a good idea to evaluate whether those tags have indeed been assigned appropriately, and, if not, possibly correct the situation. This means that we have to inspect (part of) the corpus and decide whether the tags are consistent with the tagging manual or, if the tagging manual is not clear on the subject, whether the tags have at least been applied consistently throughout the corpus. In this paper we show, by way of an experiment, how this task can be made more efficient with the help of software already in general use in wordclass tagging circles, viz. tagger generators.

The tagged corpus on which we perform our experiment consists of all the written texts of the BNC sampler CD. Its size (about 1Mw) is average for manually corrected corpora, the tagset is well-developed (C7) and the tagging process has involved the use of an equally well-developed automatic tagger (CLAWS4) and subsequent correction by a team of experienced annotators (cf. Garside and Smith (1997)). We can assume that the consistency may not be as high as that of the LOB corpus, which by now has reached an admirable level of consistency, but certainly higher than notoriously inconsistent corpora like the Wall Street Journal (cf. van Halteren et al. (To appear)).

In the following sections, we first examine the concept of consistency (section 2), then describe the tagger generator used in the experiment (section 3), evaluate the output of the experiment (sections 4 and 5), and conclude by summarising the main findings (section 6).

2 Consistency and its Evaluation

It is generally agreed that one of the desired properties of any tagging is consistency, and that we therefore want to have some means of evaluating it. An important step towards such means is an examination of what this property of "consistency" is supposed to entail, beginning with a general definition of the concept:

> When we say that somebody is *consistent*, we mean that if the same situation is encountered more than once, that person will take the same action each time.

With this general definition in place, we can take a closer look at some aspects of the concept which are important for the specific activity we are interested in, viz. the tagging of text.

First of all, we have to distinguish between *internal* consistency and consistency with regard to a *defined standard* (aka *conformance*). With wordclass tagging, there is invariably some kind of defined standard, e.g. in the form of a tagging manual. In fact, the importance of the standard is often taken to be such that deviations from it are not just called inconsistencies, but that the stronger term "errors" is used.[1] It is this type of consistency which is measured in most evaluations of the tagged material and which is referred to with "correctness" or "accuracy" percentages. However, wordclass tagging is also assumed to correspond to a general descriptive linguistic tradition (whether "theory neutral" or not), which makes it very unlikely that any tagging manual can ever really be complete. The resulting friction between the (hopefully) clear but necessarily incomplete tagging manual and each tagger's personal conception of the underlying linguistic tradition cannot but lead to individual decisions. In these cases it is impossible to evaluate the consistency with regard to the standard, as the standard is partly incomplete (the manual) and partly not well-defined (the linguistic tradition). Instead, we will have to evaluate the internal consistency, i.e. the degree to which the individual decisions have been taken consistently.

The problem with the latter kind of evaluation is that, in wordclass tagging, the concept "same situation" can be taken at different levels of granularity. When taken only in the strictest sense, it would mean that the exact same word is occurring in the exact same context.[2] It is this sense which is used when, during a tagging project, inter-annotator consistency is measured. Several taggers are given the same text and their taggings are compared. This is useful for training purposes and improvement of the tagging manual and is also a good quality control mechanism if quality is seen in relation to the manual (and possibly the more exactly defined parts of the linguistic tradition). It is not, however, very useful in the evaluation of consistency between different parts of the corpus. Barring exceptional situations, such as news items which are repeated in several broadcasts, it is extremely unlikely that there are multiple occurrences of the same word combined with the same context. This is unfortunate, as such occurrences would be extremely easy to find, and hence compare, automatically.

Internal consistency is much more likely to be expressed in terms of the same "type" of word occurring in the same "type" of context. The question, then, is if and how we can determine which types of word in which types of context are tagged differently from occurrence to occurrence. The position taken in this paper is that, just as for the tagging process itself, the best choice is a combined effort by man and machine. For the time being, only man has sufficient knowledge of the actual aims of wordclass tagging and the generalisation skills to determine which situations are indeed "the same". On the other hand, the number of situations to be examined for inconsistency is much too large for exhaustive treatment, so that some kind of sampling is necessary. Seeing that random sampling tends to reveal only the most frequent inconsistencies (see below), we will have to use the machine to select situations with a high potential for inconsistency.

Now we may not have any algorithms ready at hand which detect inconsistency, but there are quite a number of algorithms which do the opposite: machine learning algorithms are designed to try to detect consistent behaviour in order to replicate it. In the context of wordclass tagging, machine learning algorithms come in the form of tagger generators, which automatically create tagging programs on the basis of a tagged training set. If we had the ideal tagger generator and a perfectly consistent training set, the generated tagger should be able to replicate the tagging in the training set completely. This means that errors made by a generated tagger must either be due to inconsistencies in

[1] Below, we will follow this choice of terminology and use the term "error" for tags which are inconsistent with regard to the standard, leaving the term "inconsistency" for those cases where (the description of) the standard provides no information on a "correct" tag and individual choices vary.

[2] Here, we take the context to be that which a human annotator would use to make decisions. This ought to be at least the whole sentence, but might well include the surrounding paragraph or more.

the training set or to insufficiency of the learning algorithm.[3] With both causes, the situations in which errors are made can be assumed to have a high potential for inconsistency: in the first case, they are related directly to inconsistencies; in the second, they are at least non-trivial and hence possibly more error-prone for humans as well. It would therefore seem to be a good idea to focus the human evaluator's attention on those tokens for which an automatic tagger's output and the original tagging disagree.

3 Tagger Generation

For the experiment in which we test this idea, we use a new tagger generator, which is based on the Weighted Probability Distribution Voting algorithm (WPDV; cf. van Halteren (To appear)). A tagger generated by this system goes through the following steps:

1. Normally, the first step would be *tokenisation*. In our experiment, however, we use the tokenisation as present in the original tagging of the corpus, as this makes comparison much easier. This means, however, that the intelligence embedded in the tokeniser is disabled. The most important example for the data at hand is that capitalised words in headings or at the start of sentences are not decapitalised but treated as is.

   There is one area where we have to deviate from the BNC tokenization. In the sampler material, multi-token units, such as "in front of", are present as a group of tokens which together receive one tag. As we want to detect inconsistency in this grouping as well, we translate such multi-token units to sequences of separate tokens, each tagged with a ditto tag. However, as no special treatment is present for such sequences in the tagger generator, they can be expected to be responsible for a good number of errors in the tagger output.

2. Next, the *lexical lookup* component attaches to each token a list of tags which were observed with that token in the training set. Note that, as mentioned above, the token "The", e.g. at the start of a sentence, is different from the token "the".

3. For those cases where lexical lookup provides no or insufficient information, we fall back on *lexical similarity lookup*. This means that potential tags are generated by a WPDV model, using the length of the token, its pattern of character types (e.g. "1980s" would be "one or more digits followed by one or more lower case characters") and its last three actual characters. The output consists of all tags which, according to this model, are at least 0.025 times as probable as the most probable tag for the token.

4. For tokens which were observed 10 times or more in the training set, only the output of the lexical lookup is used. For all other tokens, the output of the lexical similarity lookup is added. The resulting list of tags is used in two ways. Throughout the tagging process, the full list is used as a filter on the potential tags for a token, i.e. even if the context provides overwhelming evidence that a specific tag should be used, the tag is ruled out if it does not occur in the list. Additionally, the most probable tags in the list (up to three) are used to define an *ambiguity class* (cf. Cutting et al. (1992)) for the token, which is used in the context-dependent components. The lexical probabilities of the tags are used only to determine the selection for presence (and relative position) in the ambiguity class. They are not used in the context-dependent components.

5. In the main *context-dependent components*, two WPDV models then determine the most probable tag for each token on the basis of the (disambiguated) tags of two preceding tokens and the ambiguity classes of the focus and two following tokens. The difference between the two models is that one follows the normal order of the tokens, i.e. tags from left to right, while the other uses reverse order, i.e. tags from right to left.

6. The *final selection* of the tag for each token is determined by a WPDV model using the suggestions of the two context-dependent

[3] The latter obviously in relation to the size of the training set.

models for the focus and two tokens on either side of it.

There are two reasons for the selection of this particular tagger generator. First, an evaluation with the same training and test set used by van Halteren et al. (To appear) has shown this tagging strategy to compare favourably with other state-of-the-art tagger generators: 97.82% agreement with the test set versus 97.55% for TnT (Brants, 1999), 97.52% for MXPOST (Ratnaparkhi, 1996), 97.06% for MBT (Daelemans et al., 1996) and 96.37% for the Brill tagger (Brill, 1992).[4]

Furthermore, the use of WPDV allows leave-one-out[5] application for all components[6] so that the tagger can, without any additional effort, be used in two different modes: a) with the test set equal to the training set and b) with the test set disjoint from the training set. In the first mode, the tagger will have a very large amount of specific knowledge in each situation. We should expect errors under these circumstances to show "hard" inconsistencies, such as the same word receiving different tags in the company of the same tags in the direct context. In the second mode, the tagger is operating "normally", as if tagging unseen data. Here, we should expect "soft" inconsistencies, more to do with types of words and types of contexts than with exact words and contexts. We should also expect more errors due to tagger generator learning disabilities here, and the resulting higher error rate will force us to select a smaller fraction of the errors for detailed examination.

4 Tagger-Corpus Disagreement

When a tagger is generated from the written text samples found on the BNC sampler CD, and used to re-tag those samples in the two modes described, we find an agreement rate of 99.45% when running without special measures (i.e. test equal to train) and of 96.93% when running with leave-one-out. In the first case, there are 6326 errors, in the second 35563. As we will want to compare the relative efficiency of using one run or the other, we want to examine similar numbers of errors in each case. Therefore, we take every 10th sentence for the first set (615 errors) and every 50th sentence for the second set (660 errors). Furthermore, we choose the two sets in such a way that the second set is a subset of the first one, so that we can evaluate the relative recall of the different runs. For the selected sentences, we examine all tokens where disagreement occurs.[7] In addition, in order to simulate random sampling, we take every 1000th sentence of the original corpus. For these sentences, we examine every single token (1210 tokens in total) for errors or inconsistencies in the corpus tagging, but without any reference to automatic tagger output.

Every disagreement (or observed error in the third group) is classified as to whether tagger and/or original corpus are right or wrong. Such a right-or-wrong decision is only taken if the tagging manual (or, as a backup, the linguistic tradition) is clear on the subject.[8] If such clarity does not exist, the full original corpus is inspected to determine if one of the possible tags is chosen in a substantial majority of instances of the same situation, in which case that tag is assumed to be the correct one. The resulting classification makes use of the following four classes:

T Tagger error. The original corpus is correct, the tagger is wrong.

B Benchmark error. The tagger is correct, the original corpus is wrong.

[4] These percentages have been measured on a 115Kw test set. This means that the 99% confidence intervals are 97.71–97.93%, 97.43–97.67%, 97.40–97.64%, 96.93–97.19% and 96.23–96.51% respectively.

[5] The normal way to test a tagger is by splitting the available corpus into separate training and test sets, and then train on the training set and test on the test set. In this way the test is fair, as the test data has not not been seen during training. The standard strategy is to split the corpus into 10 parts, and to repeat the train-test process 10 times, using each 10% part once as test data. This is called *10-fold cross-validation*. For some machine learning systems, however, it is possible to (virtually) remove the information about each individual instance from the model(s) when that specific instance has to be classified. This technique, called leave-one-out testing, in effect allows total cross-validation, e.g. for the case at hand one-million-fold.

[6] Even lexical lookup uses the WPDV system, so that we can use leave-one-out here as well.

[7] We ignore all other tokens. This means that, if there are tokens which receive the same erroneous tag in both original corpus and tagger output, these will not be examined, and the error will not be detected.

[8] As we are taking the point of view of the average user, we use only the tagging manual that is found on the BNC Sampler CD. No reference is made to other manuals in the CLAWS tradition, such as Johansson (1986).

Table 1: Assignment of blame for corpus-tagger disagreement (see text for key).

		T	B	X	I
full run	615	416	121	5	73
leave-one-out	660	503	84	6	67
random sample	1210	-	6	-	18

X Extreme error. Both the original corpus and the tagger are wrong.

I Inconsistency. The manual does not indicate a single correct choice and the practice in the corpus varies.

The number of times these classes are found in each of the three examinations are listed in Table 1.

Both examinations based on disagreement between automatic tagger and corpus provide a high number of inconsistency-linked situations, certainly much higher than that provided by random sample examination. Which of the two tagger runs is more useful depends on what we intend to do with the results.

The most likely aim is the identification of all erroneous tags and inconsistencies in the original corpus. In this case, we are mostly interested in recall and the leave-one-out run is preferable. Assuming that the distribution of classes remains the same throughout the corpus, examination of all 35563 disagreements found with the leave-one-out run would yield 4850 (90/660 of 35563) corpus errors and a further 3610 (67/660 of 35563) tokens which are currently tagged inconsistently and which therefore may also have to be adjusted. With the full run, we would only have to check 6326 disagreements, but this inspection would yield only 1296 errors and 751 inconsistent tags (1 in 3.7 and 1 in 4.8). We see comparable figures when we examine the part of the corpus which has been checked for both runs:[9] only 25 of the 90 corpus errors which are detected because they are flagged by the leave-one-out run are also flagged by the full run (1 in 3.6) and 15 of the 67 inconsistencies (1 in 4.5).[10] However, even the higher recall of the leave-one-out run is insufficient to find all erroneous tags and inconsistencies. In the random sample, we spotted only 6 corpus errors, but of those 6 only 2 are flagged by either tagger run, and of the 18 spotted inconsistencies, 9 escape unflagged.[11]

However, the unflagged errors and inconsistencies all show similarities in context with errors and inconsistencies which have been flagged. Therefore, we can adjust our proposal and switch to a two-phase inconsistency determination:

1. use tagger disagreement to determine contexts where inconsistency occurs
2. examine all instances of those contexts in the full corpus

With the revised strategy, recall is only interesting with regard to the number of context classes which are identified, and precision is more important, as it helps increase the efficiency of the process. Furthermore, precision is also the more important property if we do not intend to identify and correct every individual error in the corpus, but only want to get a general impression of tagging quality. From Table 1, it would seem that the full run has a higher precision, as it contains 20.5% (126/615) errors and 11.9% (73/615) inconsistencies, versus 13.6% and 10.2% for the leave-one-out run. In the next section, we will examine whether it also has sufficient recall as to the context classes we want to identify.

5 Inconsistency Context Classes

Apart from classifying who is to blame for each disagreement, we have also classified all disagreements for the type of situation they represent, i.e. their *inconsistency context class*. This classification has been done manually, and it is here that the abovementioned need for human knowledge and generalisation skills becomes very clear. As an example, where "before" in "just before the film began" is tagged II (preposition) instead of CS (subordinating conjunction), we judge that it is a case of generic preposition-conjunction confusion, and

[9] Remember that the 1/50 part of the corpus checked for the leave-one-out run is a subset of the 1/10 part checked for the full run.

[10] There is only one inconsistency flagged by the full run which is missed by the leave-one-out run. There are no errors for which this is the case.

[11] These numbers are too small for a statistically sensible extrapolation to the whole corpus.

that there is no need for subclassification based on the actual word in question or on the context. However, where the same thing happens with "as" in "such a stiff fabric as damast" we decide that this disagreement belongs to a more specific class (confusion for the word "as"), since it is the comparison aspect of "as" which leads to conjunction being preferable to preposition. The creation of classes like preposition-conjunction confusion could fairly easily be done automatically, as they correspond to specific tag (or tag group) confusions and could be based on confusion lists and numbers of times the confusion is found. However, finer distinctions like "as"-confusion, or like confusion for words ending in "-ing" when in noun-modifying position, can best be decided on manually.

The final result of the classification for the examined disagreements is a list of 51 classes, which is shown in Table 2, together with the number of corresponding disagreements in the different evaluations.[12]

For most of the classes, we find corpus errors, sometimes very unexpected ones, e.g. 4 of the 5 "single letter" errors are instances of the personal pronoun "I" which are erroneously tagged as proper noun. For some classes, only tagger errors are found, but even these may be traced back directly to corpus errors elsewhere, e.g. the 2 "letter combination" errors are both tagger errors but are clearly caused by 16 misuses of the ZZ2 tag[13] in the corpus, and the consistent mistagging by the tagger of "in front of" as preposition-noun-preposition instead of a multi-token preposition is (at least partly) due to a single such mistagging in the corpus. Only rarely, e.g. with the confusion between present tense verb and infinitive, does it appear that the blame can be put entirely on the inability of the tagger generator to learn to make the necessary distinction.

This means that practically all classes are useful for the strategy proposed above, and the tagger runs hence have to flag instances of as many classes as possible. Examination of the table shows that both the full run and the leave-one-out run provide 49 of the 51 classes. This would indicate that either run can be used, as similar numbers of inspected tokens yield similar numbers of classes. However, we would advise using a combination of the two as this is likely to provide a more varied sample. Whatever sample of flagged tokens is used, after determining the inconsistency classes, it will be necessary to use specific searches on the whole corpus to determine which words (and/or which contexts) belong to those classes.

As an example, let us look at the "preposition vs -ing participle" class. The two tagger runs only show disagreements with "including", "excluding" and "following". However, a full search shows that "barring", "concerning", "considering" and "regarding" are also tokens which are sometimes tagged as preposition and sometimes as participle. At least "concerning" and "barring" appear to have some corpus errors connected to them. The situation is especially bad for "barring", where two of the three examples are suspect: in "laws barring the manufacture of cocaine" the tag II is chosen and in "barring a disaster, the payout will be the same" the tag VVG (ing-participle).[14]

6 Conclusion

The proposed method, generating a wordclass tagger from the tagged corpus and comparing its output with the original corpus, turns out to be an efficient means of identifying inconsistency in the corpus tagging. In both modes of operation, without special measures and with leave-one-out, a substantial percentage of disagreements are linked to inconsistency.

If one intends to eradicate all errors and inconsistencies, the method will have to be combined with other types of sampling, as not all instances are themselves flagged as disagreements. However, these other types of sampling can be based on a classification of contexts underlying inconsistency. Determination of the classes involved can be done by random sampling, but is much more efficient when done on the basis of the tagging disagreements.

[12] In those cases where a disagreement could be assigned to more than one class, the most specific class has been selected, e.g. a potential location-indicating noun (NNL) in noun-modifying position is classed as special noun type rather than generic noun modifier.

[13] ZZ2 is meant for plural forms of letters, such as "a's", but is also found for tokens like "AA".

[14] In the third example, at least, the wordplay "an unusual example of a gift barring Greeks" we find the correct tag, VVG.

Furthermore, if one decides that (some of the) additional sampling is too labour-intensive,[15] inspecting and, where necessary, correcting only the flagged tokens already provides a substantial consistency improvement. Which type of run to use for this probably depends on the available manpower. The leave-one-out run provides the best recall of errors and inconsistencies, but flags about five times more tokens than the full run.

With both choices of run type, the reduction of items to be checked is dependent on the quality of the generated tagger. For the corpus and tagger generator used in this paper, the number of flagged tokens is relatively low, and certainly low enough to be manually re-checked completely. For other tagged corpora or tagger generators, the relative number may well be higher, but we expect the method to be cost-effective as long as the annotation is limited to wordclass tagging. Something which has yet to be investigated is whether the use of the same tagger generator as has been employed during the original tagging of the corpus might interfere with the inconsistency detection. While this is uncertain, it seems wise to alway use a different type of tagger generator, which should not be a problem, given the wide choice of available systems.

For other corpus annotation tasks, such as word sense tagging or syntactic annotation, the quality of machine learning systems tends to be much lower. If the automatic re-annotation method is to be used here, we strongly suggest the use of several machine learning systems. Preferably these are then combined, e.g. as described by van Halteren et al. (To appear). If the combination system is still too inaccurate for a full inspection of all flagged items, the best items to check will be those where all (or at least a substantial majority of) the systems agree, but disagree with corpus annotation. After all, a wrong prediction by one or two systems can easily be blamed on a learning disability on the part of the systems, but the same wrong prediction by a majority of the systems is a strong indication that it is probably the corpus annotation that is mistaken.

[15] E.g. there are 22900 instances of tokens which can be either preposition or conjunction.

References

J.P. Baker. 1997. Consistency and accuracy in correcting automatically tagged data. In Garside, Leech, and McEnery (eds), *Corpus annotation*, pages 243–250. Addison Wesley Longman, London.

Thorsten Brants. 1999. *Tagging and Parsing with Cascaded Markov Models – Automation of Corpus Annotation.* Saarbrücken Dissertations in Computational Linguistics and Language Technology. German Research Center for Artificial Intelligence and Saarland University, Saarbrücken, Germany.

E. Brill. 1992. A simple rule-based part-of-speech tagger. In *Proc. of the Third ACL Conference on Applied NLP, Trento.*

D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. 1992. A practical part-of-speech tagger. In *Proc. of the Third ACL Conference on Applied NLP, Trento.*

W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. 1996. MBT: A memory-based part of speech tagger generator. In Ejerhed and Dagan (eds), *Proc. of Fourth Workshop on Very Large Corpora*, pages 14–27. ACL SIGDAT.

R. Garside and N. Smith. 1997. A hybrid grammatical tagger: CLAWS4. In Garside, Leech, and McEnery (eds), *Corpus annotation*, pages 102–121. Addison Wesley Longman, London.

H. van Halteren, J. Zavrel, and W. Daelemans. To appear. Improving accuracy in NLP through combination of machine learning systems. *Computational Linguistics.*

H. van Halteren. To appear. Weighted Probability Distribution Voting, an introduction. In *Computational Linguistics in the Netherlands, 1999.*

S. Johansson. 1986. *The tagged LOB Corpus: User's Manual.* Norwegian Computing Centre for the Humanities, Bergen, Norway.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proc. of the Conference on Empirical Methods in Natural Language Processing, May 17-18, 1996, University of Pennsylvania.*

Table 2: Classes of inconsistency contexts (errors made in both corpus and tagger are shown as B and T rather than X, so that the sum of the blame types can be higher than the total count).

Context type		Full		Leave-one-out		Random	
special noun types	direction (ND)	2	(2I)	5	(5I)	-	
	title (NNA and NNB)	2	(1I 1T)	3	(1I 2T)	-	
	location (NNL)	13	(1B 12I)	20	(20I)	2	(2I)
	time (NNT)	2	(2T)	3	(2I 1T)	1	(1I)
	measure (NNU)	3	(2B 1T)	6	(3B 2I 2T)	-	
	day or month (NPD and NPM)	2	(2T)	-		-	
	capitalised word	38	(8B 5I 26T)	23	(10B 5I 12T)	4	(1B 3I)
	nominalised adjective	31	(8B 1I 21T)	19	(7B 12T)	2	(2I)
	nominalised -ing form	18	(5B 4I 9T)	17	(4B 13T)	4	(1B 3I)
noun modifiers or complements	-ing form (JJ vs VVG)	22	(3B 5I 14T)	23	(8B 5I 11T)	1	(1I)
	-ed form (JJ vs VVN)	32	(5B 9I 18T)	35	(7B 5I 23T)	1	(1I)
	-ed form of noun	1	(1T)	3	(1B 1I 1T)	-	
	-ist form (JJ vs NN)	1	(1I)	3	(3I)	-	
	capitalised word	36	(5B 5I 27T)	21	(5B 5I 12T)	3	(3I)
	other	25	(5B 4I 18T)	23	(4B 3I 16T)	1	(1B)
quantity-related	number of noun	5	(2B 1I 3T)	6	(1B 4I 1T)	-	
	quantification	16	(16T)	5	(5T)	-	
	modifier of number	12	(2B 6I 4T)	5	(2B 3T)	-	
verb tense	-ed form (past vs part)	39	(2B 1I 36T)	33	(7B 3I 23T)	-	
	base form (pres vs infin)	39	(39T)	21	(21T)	-	
	other	9	(9T)	8	(1B 7T)	-	
adverbs	adjectives used as (JJ vs R)	19	(1B 18T)	7	(2B 1I 4T)	-	
	function of (RG vs RR vs RP)	6	(6T)	4	(1B 3T)	-	
prepositions	vs conjunction	24	(5B 2I 16T)	13	(8B 5T)	-	
	vs verb particle	18	(1B 1I 16T)	21	(1B 20T)	-	
	vs locative adverb	2	(2T)	3	(3T)	-	
	vs verb participle	2	(1I 1T)	1	(1T)	-	
difficult words	as	10	(1B 9T)	12	(3B 9T)	-	
	his and her	4	(4T)	1	(1T)	-	
	once	3	(1B 2T)	-		-	
	one	7	(1B 1I 5T)	1	(1T)	1	(1B)
	’s	2	(2T)	4	(2B 2T)	1	(1B)
	so	2	(2B 1T)	3	(1B 2T)	-	
	that	6	(2B 4T)	2	(1B 1T)	-	
	there	3	(1B 2T)	1	(1B)	-	
	to	8	(1B 7T)	10	(1B 9T)	-	
	when and where	6	(1B 5T)	8	(4B 4T)	-	
not English words	capitalised foreign word	-		9	(8I 1T)	-	
	foreign word	9	(1B 3I 5T)	10	(4I 6T)	-	
	formula vs digit-letter	7	(5B 3T)	9	(7B 2T)	-	
	single letter (ZZ1)	3	(1B 2T)	5	(4B 1T)	-	
	letter combination (ZZ2)	-		2	(2T)	-	
multi-token units	unrecognised	30	(2B 2I 26T)	69	(2B 67T)	-	
	falsely recognised	17	(8B 9T)	4	(2B 2T)	2	(2I)
impossible tag for token	capitalised words	6	(1B 5T)	7	(2B 5T)	-	
	other	30	(4B 26T)	25	(8B 17T)	1	(1B)
miscellaneous	untaggable words (FU)	1	(1T)	11	(11T)	-	
	strange spelling	8	(1B 8T)	14	(5B 11T)	-	
	capitalised words	18	(18T)	14	(1B 13T)	-	
	noun-verb confusion	48	(2B 46T)	51	(7B 44T)	-	
	other	13	(13T)	12	(3B 9T)	-	

Grammar-based Corpus Annotation

Stefanie Dipper
Institut für maschinelle Sprachverarbeitung
Universität Stuttgart

1 Introduction

There is an increasing number of linguists interested in large syntactically annotated corpora (treebanks).[1] Such corpora can serve as a base for statistical applications and, at the same time, may be used in theoretical linguistics as a source for investigations about language use.

The most important treebank nowadays is the Penn Treebank (Marcus et al., 1993; Marcus et al., 1994). Many statistical taggers and parsers have been trained on this treebank, e.g. (Ramshaw and Marcus, 1995; Srinivas, 1997; Alshawi and Carter, 1994). Furthermore, context-free and unification-based grammars have been derived from the Penn Treebank (Charniak, 1996; van Genabith et al., 1999a; van Genabith et al., 1999c; van Genabith et al., 1999b). These parsers, trained or created by means of the treebank, very successfully parse unseen text with respect to correct POS tagging and chunking, and hence can be applied for enlarging the treebank.

However, the situation is different for languages other than English. Ongoing projects are still in the process of building treebanks, e.g. for German (NEGRA corpus (Skut et al., 1997), now continued in the TIGER project; the German treebank in Verbmobil (Stegmann et al., 1998)), for Czech (The Prague Dependency Treebank (Hajič, 1998)); for French (Abeillé et al., 2000). In consequence, the base that parsers could be trained on is still more or less missing. Hence alternative ways of corpus annotation that are not based on statistical parsers may be investigated.

The NEGRA/TIGER corpus consists of German newspaper texts. Currently about 30.000 sentences are annotated with dependency structures. Large parts of the annotation are performed by human annotators supported by the tool `annotate` that integrates a partial parser and a part-of-speech tagger (Brants, 2000b).

As one part of the TIGER project, it is investigated to what extent a symbolic grammar can be applied in annotation. In this approach an existing symbolic LFG grammar is used to parse the corpus. After parsing, disambiguation has to be supported manually. First results of this approach are the topic of this paper.

2 Annotation by Grammar

2.1 Scenario

In the approach presented in this paper, a broad coverage symbolic LFG grammar (Lexical Functional Grammar, (Bresnan, 1982)) is used to parse the corpus. Usually, the grammar output is ambiguous. Disambiguation is done partly manually, partly by a grammar internal ranking mechanism. Finally, the correct reading is saved in PROLOG format.

In our application, a transfer component will convert the PROLOG output into the NEGRA export format (Brants, 1997; Kuhn et al., 2000), or into other representation for-

[1] I would like to thank an anonymous referee for helpful comments on an earlier version of this paper.
The work reported here has been partially funded by the *Deutsche Forschungsgemeinschaft*, project TIGER.

mats such as an XML-based encoding format (Mengel and Lezius, 2000).

In the following sections, LFG parsing and disambiguation is presented, followed by some remarks on grammar coverage and robustness, and annotation accuracy. To illustrate these remarks, parsing results are presented in the final section.

2.2 Representations in LFG

The LFG grammar applied in parsing has been developed using the Xerox Linguistic Environment (XLE). The output of an LFG grammar basically consists of two representations, the constituent structure (c-structure) of the sentence being parsed, and its functional structure (f-structure), containing information about predicate-argument-structure, about attachment sites of adjuncts, and about tense, mood etc. In figure 1, c- and f-structure for *Maria sieht Hans* ('Maria sees Hans') are displayed.

In case of an ambiguous sentence, XLE allows for "packing" the different readings into one complex f-structure representation. All features are represented only once; feature constraints that only hold in one of the readings are marked by variables. The result is an f-structure that is annotated with variables to show where alternatives are possible.

In figure 2, the alternative c-structures for *Maria sieht Hans mit dem Fernglas* ('Maria sees Hans with the telescope') are displayed. The readings differ with respect to the attachment site of the PP *mit dem Fernglas*, either dominated by VP or by NP.

Figure 3 shows the corresponding f-structures, combined in a single f-structure. The PP's f-structure, embedded under the feature ADJUNCT, is displayed only once. In the example, variables a:1 and a:2 indicate the alternative attachments.

The correct reading is selected by a human annotator after parsing. Selection is done either by picking the correct c-structure tree or by clicking on the respective variables in the f-structure.[2]

2.3 Semi-automatic Disambiguation

In the scenario sketched above, disambiguation is exclusively done by a human annotator. In fact, however, XLE provides a (non-statistical) mechanism for suppressing certain ambiguities automatically. The mechanism consists of a constraint ranking scheme inspired by Optimality Theory (OT) (Frank et al., 1998). Each rule and each lexicon entry can be marked by so-called OT marks. When a sentence is parsed, each analysis is annotated by a multi-set of OT marks. The OT marks keep a record of all rules and lexicon entries being used during the parse to arrive at the analysis in question. The grammar contains a ranked list of all OT marks. When an ambiguous sentence is parsed, the OT mark multi-sets of all readings compete with each other. A multi-set containing a higher ranked OT mark than another multi-set is filtered out.

An example is given in (1). In German, the subject as well as the object can occupy the first position (1a,b). If neither the subject nor the object is overtly case marked, both readings are possible in principle (1c). But in fact, the order subject – object is far more frequent. Hence the second reading can be suppressed by an OT mark. Note that this does not generally exclude objects in first position – as soon as objects are case-marked in an unambiguous way, they are not suppressed any more.

(1) a. der Hans sieht Maria.
the(nom.) H. sees M.
'Hans sees Mary.'

b. den Hans sieht Maria.
the(acc.) H. sees M.
'It is Hans that Mary sees.'

c. Hans sieht Maria.
H. sees M.
'Hans sees Mary.' (preferred)
'It is Hans that Mary sees.'

[2] XLE provides various browsing tools applying to c-structure as well as to f-structure which can be used for manual disambiguation (cf. (King et al., 2000) where these tools are described extensively). This is similar to the syntactic and semantic sentence properties that are displayed by the disambiguation tool "TreeBanker" (Carter, 1997).

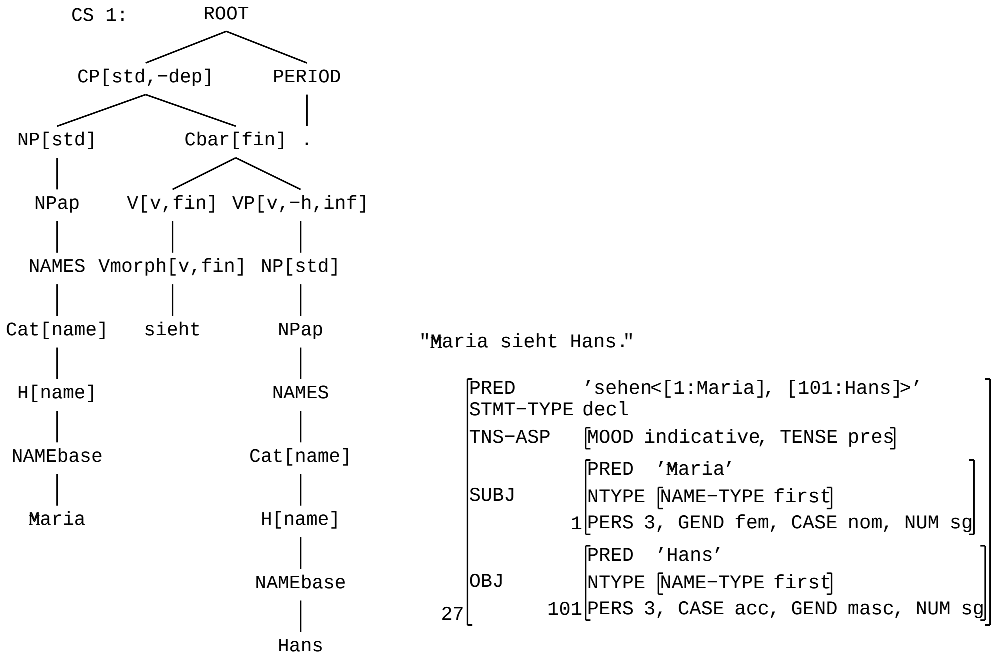


Figure 1: c- and f-structure for *Maria sieht Hans*

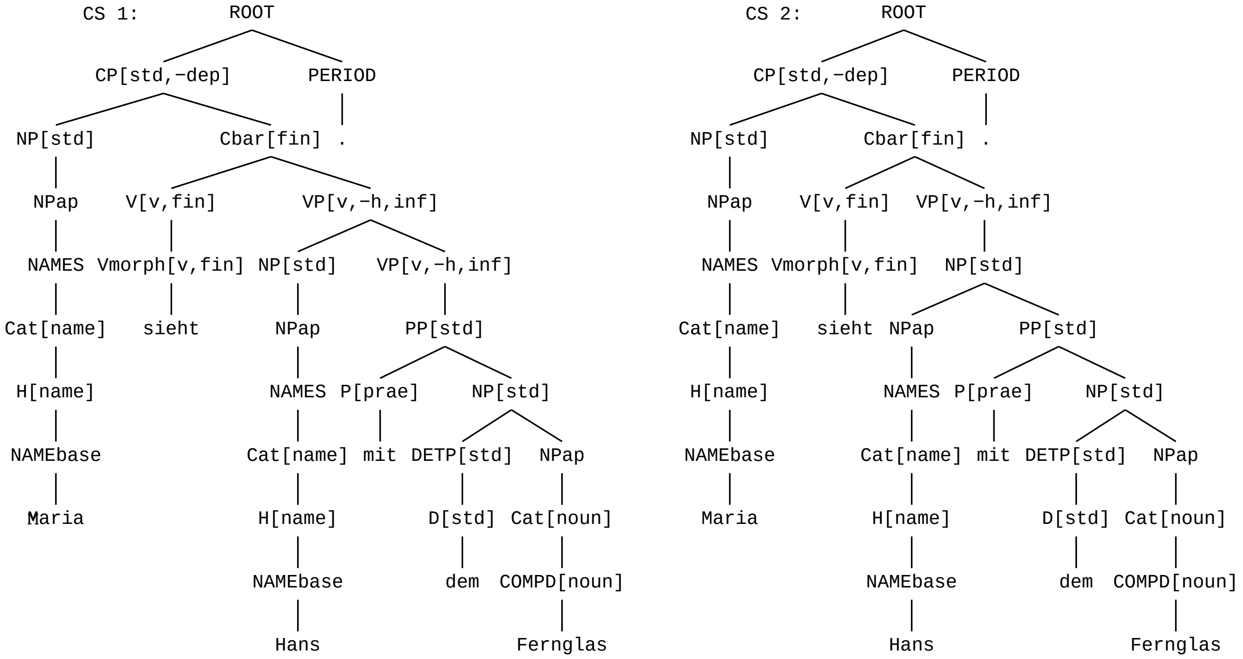
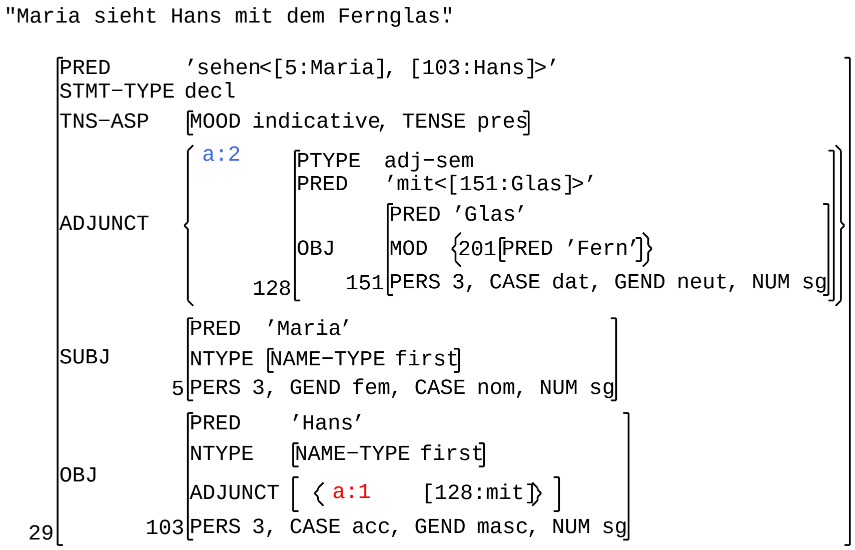


Figure 2: c-structures for *Maria sieht Hans mit dem Fernglas*

In those cases where the correct reading is erroneously suppressed (if, for example, the correct reading does have an object without case-marking in first position), the relevant OT mark can easily be deactivated by the human annotator.

In the ambiguous example presented in 2.2, two readings in fact have been sup-



```
"Maria sieht Hans mit dem Fernglas."

   [PRED       'sehen<[5:Maria], [103:Hans]>'                                  ]
   |STMT-TYPE decl                                                             |
   |TNS-ASP    [MOOD indicative, TENSE pres]                                   |
   |           { a:2    [PTYPE  adj-sem                               ]}       |
   |           |        |PRED   'mit<[151:Glas]>'                     ||       |
   |ADJUNCT    |        |        [PRED 'Glas'                         ||       |
   |           |        |OBJ     |MOOD {201[PRED 'Fern']}            ||       |
   |           |     128[     151[PERS 3, CASE dat, GEND neut, NUM sg]]}       |
   |           [PRED  'Maria'                       ]                         |
   |SUBJ       |NTYPE [NAME-TYPE first]              |                         |
   |          5[PERS 3, GEND fem, CASE nom, NUM sg]                           |
   |           [PRED     'Hans'                       ]                       |
   |           |NTYPE    [NAME-TYPE first]            |                       |
   |OBJ        |ADJUNCT  [ { a:1    [128:mit]} ]      |                       |
 29[        103[PERS 3, CASE acc, GEND masc, NUM sg]                          ]
```

Figure 3: Packed f-structure for *Maria sieht Hans mit dem Fernglas*

pressed by this mechanism. Without OT marks, the f-structure for *Maria sieht Hans mit dem Fernglas* contains two additional analyses with *Hans* as subject, cf. figure 4.

Very often, however, the OT mechanism does not help to determine the correct reading, e.g., when adverb attachment is involved. In these cases, the parser outputs all remaining readings, and disambiguation has to be done manually.[3]

2.4 Coverage and Robustness

For building large annotated corpora, consecutive sentences have to be parsed. Thus, coverage and robustness of the grammar used for annotation is important.

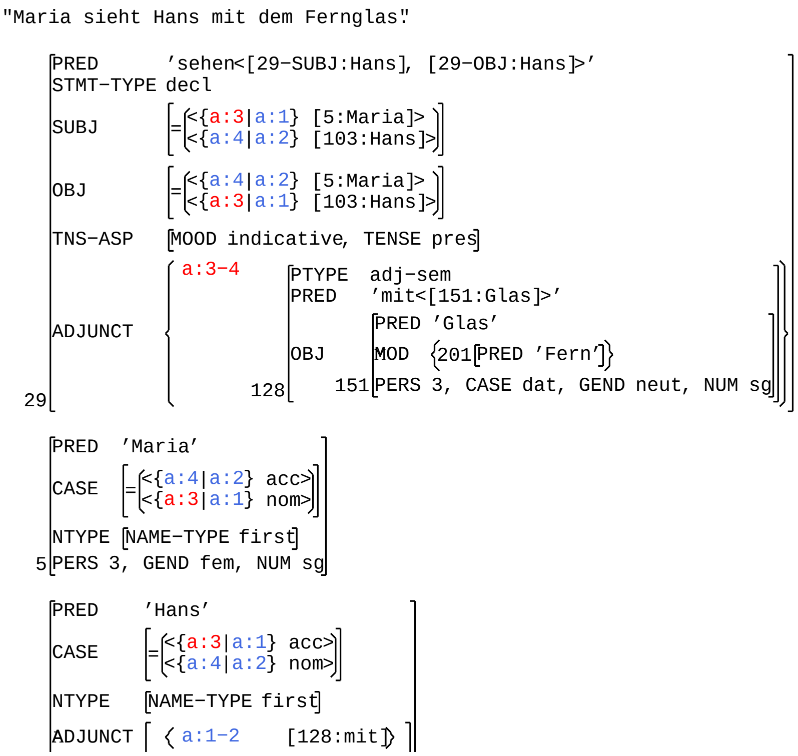
Statistical approaches clearly cope better with free, random text than symbolic approaches. On the one hand, statistical taggers and parsers are able to analyze defective input such as sentences containing typing errors or even ungrammatical sentences. On the other hand, they can provide analyses for rare constructions without getting into ambiguity problems when parsing ordinary sentences – in these cases, rare construction rules are suppressed automatically.

In contrast, parsing by a pure (i.e. non-statistical) LFG grammar yields deep and detailed analyses but at the cost of lower coverage and robustness. Purely symbolic parsing therefore requires text preprocessing.[4] Typing errors and other shortcomings must be corrected, special constructions like newspaper headers have to be marked. For an optimal result, proper nouns such as names of people, organizations, etc. should be listed in a lexicon.

However, even after the best possible text preprocessing and lexicon completion, there will certainly still be constructions that are not parsed by the grammar, e.g., constructions like ellipses and non-constituent coor-

[3] In (Riezler et al., 2000), a statistical model applied to an LFG grammar for German is presented that may be used to support manual disambiguation.

[4] Especially in the domain of speech data processing, much research has been devoted to robust parsers. (Rosé and Lavie, To appear) show that even with a symbolic LFG-style grammar, the parser's flexibility can be increased to cope with word skipping, insertions, etc. However, since this increases the amount of ambiguity, a statistical disambiguation is a prerequisite – which we do not have currently.



```
"Maria sieht Hans mit dem Fernglas."

    PRED       'sehen<[29-SUBJ:Hans], [29-OBJ:Hans]>'
    STMT-TYPE  decl
    SUBJ       =<{a:3|a:1} [5:Maria]>
                <{a:4|a:2} [103:Hans]>
    OBJ        =<{a:4|a:2} [5:Maria]>
                <{a:3|a:1} [103:Hans]>
    TNS-ASP    [MOOD indicative, TENSE pres]
                 a:3-4
                         PTYPE  adj-sem
                         PRED   'mit<[151:Glas]>'
    ADJUNCT    {                PRED 'Glas'
                         OBJ    MOD  {201[PRED 'Fern']}
                      128    151PERS 3, CASE dat, GEND neut, NUM sg
  29

    PRED   'Maria'
    CASE   =<{a:4|a:2} acc>
            <{a:3|a:1} nom>
    NTYPE  [NAME-TYPE first]
   5PERS 3, GEND fem, NUM sg

    PRED     'Hans'
    CASE     =<{a:3|a:1} acc>
              <{a:4|a:2} nom>
    NTYPE    [NAME-TYPE first]
    ADJUNCT  [ { a:1-2     [128:mit]}
```

Figure 4: f-structure for *Maria sieht Hans mit dem Fernglas*, no OT filtering

dination. These constructions are problematic since adding the respective rules raises the number of (unwanted) ambiguities for nearly all sentences, and, in addition, it has a negative impact on parsing efficiency.[5]

Clearly, bad coverage and robustness is a problem for grammar-based corpora annotation. XLE provides a special mechanism to improve coverage and robustness. Certain rules or restrictions can be marked by special "STOPPOINT" OT marks. If a sentence is now parsed, these rules or restrictions are ignored. Only if the first parse fails are these rules or restrictions activated and a second parse is started. In this way, rules for rare constructions can be added and restrictions (for instance, on agreement) can be relaxed, without causing serious ambiguity problems for ordinary sentences. Currently we use STOPPOINT OT marks for verb participles used adverbially or in copula constructions. Many of them actually are lexicalized (like *dringend* 'urgent', *verrückt* 'crazy') but nevertheless may be missing from our lexicon. Hence we allow for these participles in general in a second parse, without getting additional readings for each sentence in analytic past tense, i.e. containing an auxiliary plus participle. Further research has to show how to apply this mechanism in an optimized way.

2.5 Accuracy

With respect to accuracy, a grammar-based annotation performs well. We mention three aspects of the approach presented here that support accuracy of annotation.

First, an analysis by an LFG grammar is syntactically consistent, otherwise the parse would have failed. For example, LFG analyses never contain inconsistencies such as the following: missing subject-verb agreement; words tagged as infinitive but functioning as the head of a finite clause; the head of a NP tagged as nominative but the NP function-

[5] Note that non-constituent coordination can be handled in LFG in an elegant way (Maxwell and Manning, 1996). So again, the problem is one of ambiguity management.

ing as an accusative object; etc.

Second, the grammar certainly is not error-free and grammar internal errors may carry over to the analyses but these errors are systematic. If, for example, a proper noun like *Kohl* is not listed as a name in the grammar's lexicon, all analyses of sentences about the person Helmut Kohl falsely contain the reading of *Kohl* as a common noun ('cabbage'). But once the error is detected in one analysis or in the grammar itself, it is often possible to automatically track down all other instances of the same error occurring previously in the annotation. Note, however, that such errors may be difficult to detect.

Third, manual disambiguation of LFG analyses usually does not impair accuracy of the annotated corpus, since in many cases, disambiguation is guided by prominent properties. When picking the correct reading, the human annotator can make use of clear, prominent properties of the analyses, namely constituent structure and predicate-argument-structure.

2.6 Some Performance Data

To illustrate the findings of the preceding sections, we present some figures indicating the grammar's performance. Note, however, that the grammar has not been tuned or trained with respect to the corpus.

In a first experiment, 2000 sentences from the TIGER corpus (German newspaper texts) were parsed. In a first pass, the text was parsed without any preprocessing (except for splitting the text into sentences). In a second pass, header markers were added and quotes were removed (since the grammar currently does not accept quoted text; the quotes can be easily recovered after parsing).[6] These text modifications were done automatically. The grammar performance improved considerably, cf. rows 1 and 2 in figure 5.

Then the grammar was partly rewritten with two major modifications: first, the grammar was tuned for efficiency (without affecting coverage); second, PP and adverb attachment were allowed in a more general way than in the previous grammar version. This increased coverage as well as ambiguity, as can be seen in the third row, reporting about 6000 sentences (preprocessed in the same way as in the second pass).

The first column shows the number of sentences in the test corpus, the second column shows the number of sentences that got a parse (without checking for correctness). As can be seen, in the first pass only 28% of the sentences were parsed as opposed to 40% after some text preprocessing. After some general grammar modifications, 47% were parsed.[7]

The third column contains the number of analyses or readings per parsed sentence. Only readings that were not filtered out by the XLE internal disambiguation mechanism are taken into account (hence "optimals"). Both average as well as median are given. As can be seen from figure 5, in the third pass the average number of readings increased massively. But nevertheless the median is 2, so most of the sentences are still easy to disambiguate manually. Note that in this experiment, it was not checked whether the correct reading was among the analyses.

The forth column reports about the number of analyses that were suppressed by XLE disambiguation (hence "suboptimals").

Finally, average parsing time and number of tokens per sentence are given.

In a second experiment, 300 sentences were parsed and the analyses were evaluated.

[6] Quotes are problematic for several reasons: They are ambiguous and either mark direct speech or quote material in the running text. Quotes do not always correspond to constituents boundaries and matching pairs of quotes may be distributed over distinct sentences.

[7] We are only aware of one sentence-based evaluation involving a grammar with comparably deep analyses: without tuning, the XTAG grammar parsed 39.09% of 6364 sentences ($\leq$ 15 words long) from the Wall Street Journal with an average of 7.53 analyses per sentence (Doran et al., 1994). Other evaluations usually measure performance below sentence level, such as chunking or (super)-tagging (Srinivas, 1997; Ramshaw and Marcus, 1995; Brants, 1999), and hence are not comparable with our grammar that does not yield partial analyses (yet).

	#sentences	parsed	optimals		suboptimals		time(sec)		#tokens
			Ø	Med	Ø	Med	Ø	Med	Ø
1.	2000	553 (= 28%)	7	2	1689	7	17	1.8	15.5
2.	2000	809 (= 40%)	6	2	3480	10	17	1.8	15.3
3.	6000	2833 (= 47%)	28	2	34331	18	14	1.9	16.0

Figure 5: LFG parsing results for German newspaper sentences

160 sentences were parsed by the grammar; among these, 120 parses contained the correct reading (the correct reading had to be part of the "optimal" analyses), cf. figure 6.

We also did some preliminary evaluation of the errors.

- 10% of the sentences were not parsed because of gaps in the morphological analyzer.[8]
- 4% of the sentences failed because of storage overflow or timeouts (with limits set to 100 MB storage and 100 seconds parsing time).
- More than 30% of the sentences failed because gaps in the lexicon, which are mostly due to missing subcategorization frames.[9]

We decided not to manually disambiguate sentences that get more than 20 analyses. This is the case for 5.8% of the sentences. With this restriction, a trained human annotator disambiguates about one sentence per minute on average.[10]

To sum up the findings of this section: in the short-term, these data suggest the necessity of the following: further text preprocessing such as correction of typing errors; completion of the grammar's lexicon by extracting unknown words from the corpus.

However, in the long-term, we will have to apply statistical disambiguation. This will allow us to include robustness mechanisms.

In the meantime, the remainder of the sentences that have not been correctly parsed by our grammar are annotated by means of the tool `annotate`.

3 Conclusion and Outlook

We have presented first results in syntactic annotation of a large German corpus by a symbolic LFG grammar. On average, the grammar parses 47% of the sentences. Among these, 75% contain the correct reading. Disambiguation is done partly by the XLE internal ranking mechanism. Remaining ambiguities (median: 2) are solved by a human annotator. This takes about one minute per sentence with an average length of 16.0 tokens.

By means of a transfer component, LFG representations can be converted into canon-

[8] We use a guesser mechanism for capitalized words that also handles genitive and plural inflection. All morphological failures are due to non-capitalized unknown words or else capitalized words containing strings other than characters or numbers.

[9] The base lexicon is mainly extracted automatically from corpora (Eckle-Kohler, 1999) and mostly consists of subcategorization frames (in the TSNLP format). There are 14.000 verb lemmata with 28.500 frames (115 different frames); 1100 adjective lemmata with 1650 frames (17 different); 780 noun lemmata with 970 frames (3 different). The TSNLP frames are converted automatically into an LFG format (Bröker and Dipper, 1999).

[10] This result is very similar to that reported in (Brants, 2000a), where a trained annotator needs on average 50 seconds to annotate a sentence with an average length of 17.5 tokens.

#sentences	parsed	correct reading among optimals
300	160 (= 53%)	120 (= 40%)

Figure 6: Evaluation of 300 sentences

ical treebank formats.

Coverage and robustness are weak points in grammar-based annotation. The performance data presented in 2.6 point to a need to further exploit text preprocessing and to complete the grammar's lexicon. In the longer term, however, statistical disambiguation and robustness mechanisms such as relaxation of certain restrictions have to be investigated.

References

Anne Abeillé, Lionel Clément, and Alexandra Kinyon. 2000. Building a treebank for French. In *Proceedings of the LREC 2000*, Athens, Greece.

Hiyan Alshawi and David Carter. 1994. Training and scaling preference functions for disambiguation. *Computational Linguistics*, 20(4).

Thorsten Brants. 1997. The NeGra export format for annotated corpora (version 3). Technical report, NEGRA Project, Universität des Saarlandes.

Thorsten Brants. 1999. Cascaded Markov Models. In *Proceedings of 9th Conference of the EACL 1999*, Bergen, Norway.

Thorsten Brants. 2000a. Inter-annotator agreement for a German newspaper corpus. In *Proceedings of the LREC 2000*, Athens, Greece.

Thorsten Brants. 2000b. Interactive corpus annotation. In *Proceedings of the LREC 2000, Athens, Greece.*

Joan Bresnan, editor. 1982. *The Mental Representation of Grammatical Relations.* MIT Press.

Norbert Bröker and Stefanie Dipper. 1999. Zur Konstruktion von Lexika für die maschinelle syntaktische Analyse. In J. Gippert and P. Olivier, editors, *Multilinguale Corpora - Codierung, Strukturierung, Analyse. 11. Jahrestagung der Gesellschaft fuer Linguistische DatenVerarbeitung.* Enigma corporation, Prag.

David Carter. 1997. The TreeBanker: a tool for supervised training of parsed corpora. In *Proceedings of the ACL Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, Madrid, Spain.

Eugene Charniak. 1996. Tree-bank grammars. In *AAAI-96. Proceedings of the Thirteenth National Conference on Artificial Intelligence.* MIT Press.

Christine Doran, Dania Egedi, Beth Ann Hockey, B. Srinivas, and Martin Zaidel. 1994. XTAG system – a wide coverage grammar for English. In *Proceedings of International Conference on Computational Linguistics (COLING 94)*, Kyoto, Japan.

Judith Eckle-Kohler. 1999. *Linguistisches Wissen zur automatischen Lexikon-Akquisition aus deutschen Textkorpora.* Logos, Berlin.

Anette Frank, Tracy Holloway King, Jonas Kuhn, and John Maxwell. 1998. Optimality theory style constraint ranking in large-scale LFG grammars. In *Proceedings of the LFG98 Conference*, Brisbane, Australia. CSLI Online Publications, http://www-csli.stanford.edu/publications.

Jan Hajič. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In Eva Hajičova, editor, *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová.* Charles University Press, Prag.

Tracy Holloway King, Stefanie Dipper, Anette Frank, Jonas Kuhn, and John Maxwell. 2000. Ambiguity management in grammar writing. In *Proceedings of the ESSLLI 2000 Workshop on Linguistic Theory and Grammar Implementation*, Birmingham, Great Britain.

Jonas Kuhn, Heike Zinsmeister, and Martin Emele. 2000. From LFG structures to TIGER treebank annotations. Presented at the Workshop on Syntactic Annotation of Electronic Corpora, University of Tübingen.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2).

Mitchell P. Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the Human Language Technology Workshop*. Morgan Kaufmann.

John T. Maxwell and Christopher D. Manning. 1996. A theory of non-constituent coordination based on finite-state rules. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG96 Conference*, Grenoble, France. CSLI Online Publications, http://www-csli.stanford.edu/publications.

Andreas Mengel and Wolfgang Lezius. 2000. An XML-based encoding format for syntactically annotated corpora. In *Proceedings of the LREC 2000*, Athens, Greece.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, Dublin, Ireland.

Stefan Riezler, Detlef Prescher, Jonas Kuhn, and Mark Johnson. 2000. Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Proceedings of the ACL 2000*, Hong Kong, China.

Carolyn Penstein Rosé and Alon Lavie. To appear. Balancing robustness and efficiency in unification-augmented context-free parsers for large practical applications. In van Noord and Junqua, editors, *Robustness in Language and Speech Technology*. Kluwer Academic Press.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of ANLP-97*, Washington.

B. Srinivas. 1997. Performance evaluation of supertagging for partial parsing. In *Proceedings of Fifth International Workshop on Parsing Technology*, Boston, USA.

Rosmary Stegmann, Heike Schulz, and Erhard Hinrichs. 1998. Stylebook for the German treebank in Verbmobil. Verbmobil, Universität Tübingen.

Joseph van Genabith, Louisa Sadler, and Andy Way. 1999a. Data-driven compilation of LFG semantic forms. In *Proceedings of the EACL 1999 Workshop on Linguistically Interpreted Corpora (LINC-99)*, Bergen, Norway.

Joseph van Genabith, Louisa Sadler, and Andy Way. 1999b. Semi-automatic generation of f-structures from treebanks. In *Proceedings of the LFG99 Conference*, Manchester, Great Britain. CSLI Online Publications, http://www-csli.stanford.edu/publications.

Joseph van Genabith, Louisa Sadler, and Andy Way. 1999c. Structure preserving CF-PSG compaction, LFG and treebanks. In *Proceedings of the ATALA Treebank Workshop*, Paris, France.

Automatic procedures in tectogrammatical tagging

Alena BÖHMOVÁ
ÚFAL MFF UK
Malostranské nám. 25
118 00 Prague, Czech Rep.
bohmova@ufal.mff.cuni.cz

Petr SGALL
ÚFAL MFF UK
Malostranské nám. 25
118 00 Prague, Czech Rep.
sgall@ufal.mff.cuni.cz

Abstract

This paper describes a specific part of the Prague Dependency Treebank annotation, the step from the surface dependency structure towards the underlying representation of the sentence. The first section explains the theoretical basis of the project. In Section 2 all the procedure of conversion to the tectogrammatical structure is summarized and Section 3 presents in detail the present stage of the automated part of the conversion procedure.

1 Introduction

A semi-automatic syntactic annotation of a part of the Czech National Corpus in the Prague Dependency Treebank (PDT) has among its aims the possibility to check the theoretical approach chosen (Functional Generative Description, see [2]), as well as to establish a basis that could serve as a suitable starting point for a large-scale monographic analysis of the numerous problems of the sentence structure in general and of the grammar of Czech in paritcular which still require empirical research. Such an analysis is expected to be helpful at least in three respects:

(i) to make a relatively complete description of Czech into a realistic task,

(ii) to fill lacunes present in the prelimnary form of the annotation procedures formulated up to now, and

(iii) to proceed towards a procedure that would be automatized to a maximally high degree.

We do hope that our paper may be useful in attracting more attention not only to the need of an annotation reaching the underlying sentence structure (rather than just the usual ’surface-structure’ parsers, which may help in natural language processing, although not that much in achieving the aims of a theoretical linguistic description), but also to a deep-reaching comparison of different approaches to syntax. We are convinced that dependency-based syntax with its maximally economical tree structures may be of particular interest for the aims of contemporary computational linguistics. This holds especially of an approach in which function words are classed together with inflectional morphemes as corresponding to indices in complex node labels, rather than to specific nodes, and in which also other aspects of the underlying, tectogrammatical, structure are established. Moreover, a comparison of the problems concerned in syntactic annotation procedures for languages of different types certainly can be important if general theories of language description are looked for and compared with each other; such a theory should show the core of linguistic structure to be economical enough both to help explain the easiness of mother tongue acquisition and to be implementable in computers.

A language with rich inflection and with a high degree of "free" word order, such as Czech, cannot be handled by primarily using cues based on cooccurrence with neighboring items, but requires specific procedures for the disambiguation of morphemic units (prepositional and simple case forms, agreement forms, etc.), which hardly could be fully automated. The work on such procedures has led to our conviction that many insights of classical structural linguistics may still be highly useful, although they have not been duly reflected in theories using an approach based on constituency (that originated with Bloomfieldian descriptivism). Considering syntactic dependency (which is being developed on the basis of the work of L. Tesnière) to constitute the primary layer of sentence patterns, we work with a structure that corresponds to extremely flat constituency patterns, and we use no nonterminals in the dependency trees. Instead of notions such as NP or AP, the dependency approach shows just items dependent (immediately or not) on a noun or an adjective, respectively. A detailed discussion of tectogrammatics, which cannot be included into the present paper, can be found in [2], [4].

The following strategy of annotation has been found useful, and this may hold also for many other languages: The first phases of the annotation of PDT are (i) the morphemic representations and (ii) the dependency trees on an intermediate (’surface’) analytic level, i.e. analytic tree structures (ATSs, see [1]), where (i) has to use a combination of statistical and structural methods to obtain a reliable automatic treatment, and (ii) has to be carried out manually. While (i) and (ii) have been discussed elsewhere, the present paper is devoted to a subsequent phase (iii), the transduction (conversion) from ATSs to (underlying) syntax itself, i.e. to tectogrammatical representations, which should be provided for 10 000 sentences during the year 2000 (at its start, 100 000 sentences have obtained their ATS annotations).

The main points of the transduction include:

(a) deleting those nodes of the ATSs which correspond to function words and to most punctuation marks, with an indication of their functions in the form of indices of the corresponding lexical (i.e. autosemantic, rather than auxiliary) occurences; as an exception, we use nodes for coordinating conjunctions (as heads of the coordinated constructions), thus working with underlying representations in the specific form of ’tectogrammatical tree structures’ (TGTSs); (b) assigning every lexical occurrence the appropriate syntactic functors (which distinguish more than 40 kinds of syntactic relations, i.e. of kinds of valency slots, e.g. PAT (patient or objective), ADDRessee, LOCative, MANNer) and morphological grammatemes (marking the values of tense, aspect, modalities, number, etc.), as well as syntactic grammatemes (values such as ’in, on, under, among’ with Locative or Directional);

(c) restoring those nodes of TGTSs which are deleted in the surface form of the input sentences;

(d) indicating the position of every node in the topic-focus articulation (TFA) with a scale of communicative dynamism, represented as underlying word order (see [2], [3] for a discussion of TFA).

2 Automatic parts of transduction:

The transduction from ATSs to underlying trees has the following three parts, the first of which is discussed in more detail in Section 3:

(i) an automatic ’pre-processing’ module,

(ii) a manual part, which changes the analytic functions (esp. Subject, Object, Adverbial, Attribute), into corresponding functors (only the most basic cases are changed automatically); nodes for the deleted items are ’restored’ (mostly as pronouns); the TFA indices for focus, contrastive and non-contrastive topic are specified; a ’user-friendly’ software enables the annotators to work with diagrammatic shapes of trees;

(iii) a subsequent automatic module adds first of all

(a) information on the lexical values of restored nodes in unmarked cases in which the (marked) values have not been specified in (ii): esp. in coordinated constructions the values of the (symmetric) counterparts in the given construction are added;

(b) certain values of syntactic grammatemes (esp. where a preposition allows for a reliable choice);

(c) at the same time, the gender and number values are cancelled whenever they only indicate agreement (as with adjectives in most positions), and

(d) the remaining nodes corresponding to commas, dashes, quotes, etc. are deleted.

In the next months, the automatic procedure is supposed to be enriched in various respects, such as the build-up of the lexicon (with entries including the valency frames), word derivation, and the degrees of activation of the ’stock of shared knowledge,’ as far as derivable from the use of nouns and pronouns in subsequent utterances. Several types of grammatical information, e.g., the disambiguated values of prepositions and conjunctions, can only be specified after further empirical investigations, in which, whenever possible, also statistical methods will be used. In any case, the annotated corpus will offer a suitable starting point for monographic elaboration of the problems concerned.

3 The first part of the automatic transduction

3.1 TGTS description

Every node of the TGTS contains all the information inherited from the ATS, and new attributes are added.

The `trlemma` attribute contains the lemma of the node. The `trlemma` of a single node (even if the node is hidden, i.e. marked as absent in the TGTS) is equal to its analytical lemma assigned in the ATS. The compound nodes that represent more than one word of the surface sentence are assigned the `trlemma` attribute in the following way:

- Verbal nodes: lemma of the autosemantic (lexical) verb.
- Compound prepositions, conjunctions and numeratives: `trlemma` is composed of the lemmas of the parts of the item (e.g. the three nodes representing numerative 1150 ‘tisíc sto padesát’ are joined into one node with `trlemma` = ‘tisíc_sto_padesát’).
- Newly added nodes are assigned either proper lexical values (in case of filled deletions - mostly pronouns), or technical lexical values, such as ‘Gen’ for the general participant, ‘Cor’ for the coreferential node of a controlee, or ‘Neg’ for negation.

The morphological grammatemes are captured using the attributes of: gender, number, degree of comparison, tense, aspect, iterativeness, verbal modality, deontic modality, sentence modality.

Next to the morphological grammatemes there are attributes describing the position of the node at the tectogrammatical level: topic-focus articulation, functor, syntactic grammateme, type of relation (dependency, coordination, apposition), phraseme, deletion, quoted word, direct speech, coreference, antecedent and some other, technical attributes. The attribute ‘function word (`fw`)’ is used for storing the preposition or conjunction of the word for the later resolution of the syntactical grammatemes. The attributes ‘deep order (`dord`)’ and ‘sentence order (`sentord`)’ are used to distinguish between the sentence surface word order and the deep word order.

3.2 The steps of the procedure

3.2.1 Auxiliary verbs, i.e. `verbmod` *attribute*

The verb is conjoined with its auxiliary nodes into a complex value of a single node, placed in the highest position in the relevant subtree. All AuxV nodes are hidden. The verb is assigned the values of the grammatemes of tense and verb modality on the basis of the lexical values of these auxiliary nodes. The lemma of the autosemantic verb is put into the trlemma attribute of the remaining node, which is assigned the grammateme values depending on the AuxV dependent nodes.

The tables below show what assignments are made in the automatic procedure for the verbal node. Table 1 contains the rules applied to the nodes for autosemantic verbs, the rules are captured in the table rows in the sequence they are being used. If all the conditions are fulfilled for some node, the rule is applied. E.g. the second row of the table reads as follows: If the verb daughter node is labelled either with the lemma “být” or“by”, disregarding the possible presence of “se” (which was already handled by rule 1), and the morphological tag of the verb begins “VR” (symbol for preterite tense), then assign the verb attribute `tense` the value ANT.

	Presence of dependent node with lemma			Morph. tag of the verb	Assigned attributes
no	**být** **(to be)**	**by** **(cond.)**	**se, f=AuxT**		
1	-	-	yes	-	trlemma => attach '_se' to the trlemma of the verb
2	no	no	-	VR	tense => ANT
3	no	no	-	VU	tense => POST
4	no	no	-	other	tense => SIM
5	no	yes	-	-	tense => SIM verbmod => CDN
6	yes	yes	-	-	tense => ANT verbmod => CDN
7	yes	no	-	-	tense => ANT

Table 1. *Verbs*

Examples:

(i) **otevřel**.VR **se**.AuxT =>
trlemma=**otevřít_se** (rule 1)
tense=**ANT** (rule 2)
E: *(it) opened*

(ii) **učil**.VR **by**.AuxV **se**.AuxT =>
trlemma=**učit_se** (rule 1)
tense=**SIM,** verbmod=**CDN** (rule 5)
E: *(he) would learn*

(iii) **byl**.AuxV **by**.AuxV **spal**.VR =>
trlemma**=spát**
tense=**ANT**, verbmod=**CDN** (rule 6)
E: *(he) would have slept*

(iv) mohla jsem být (já) spatřena
trlemma**=spatřit**
tense=**ANT** (rule 7)
deontmod=**POSS**

E: *(I) could have been seen*

3.2.2 Modal verbs, i.e. deontmod *attribute*

The modal verb is merged with the autosemantic verb depending on it in the ATS. The transduction procedure consists in three steps: the tree is rearranged in that the modal verb depends on the autosemantic verb, the value for the attribute deontmod of the latter verb is assigned its value according to the lexical value of the modal verb, and the modal verb node is deleted.

Modal verb	**English transl.**	**Auto-semantic verb form**	**f of the verb**	deontmod **assigned**
chtít	want	infinitive	object	VOL
muset	must	-		DEB
moci, dát_se	can	-		POSS
smět	be allowed	-		PERM
umět, dovést	can	infinitive	object	FAC
mít	should	infinitive	object	HRT

Table 2. Modal verbs.

3.2.3 Prepositions and conjunctions, i.e. fw *attribute*

Every preposition node is deleted and its lexical value is stored in the attribute fw of the noun. The preposition will be used for the future (at least partly automatized) determination of the value of the syntactic grammateme of the noun.

Every subordinating conjunction node is deleted. Its lexical value is stored in the fw attribute of the head verb of the subordinate clause. Conjunctions for coordination and apposition are used in the tectogrammatical tree as the heads of the coordinated clauses.

3.2.4 General actor

The reflexive particle 'se' has three possible analytical functions in a Czech sentence. The analytical function value AuxT is assigned to a reflexive 'se' having the function of lexical derivation (of a middle verb). As shown in Table 1, 'se' is conjoined with the lemma of the verb in such case. If 'se' was assigned the function 'AuxR' at the analytical level, it expresses a general actor of the verb. The node is preserved, its attribute trlemma is filled with the 'Gen' value and its functor is 'ACT'. If 'se' was assigned the function 'OBJ', it gets the functor 'PAT'.

3.2.5 Quotation marks, i.e. `quot` *attribute*

The sentence is searched for quotation marks. If a whole clause having the form of a sentence is inserted into a pair of double quotes, its verb obtains the value ’DSP’ (direct speech) on the attribute `quot`. If only one token of a double quote appears in the sentence, the attribute `quot` of the head word(s) of the string containing the quote is assigned ‘DSPP’ value (direct speech part). Otherwise, the head word(s) of the string enclosed in quotes is/are assigned `quot` = ’QUOT’ (quoted word).

3.2.6 Punctuation

All punctuation nodes (which have the analytical function ‘AuxX’) are hidden except for the following two cases:

- the node for a comma placed in the position directly following a noun is left in the tree to enable the annotators to decide about the type of the adjunct (restrictive or descriptive),
- a comma node that is a bearer of coordination or apposition is not deleted, as far as this function can be recognized from the ATS.

The `trlemma` attribute of undeleted comma node is filled with `Comma` value.

3.2.7 Node for negation

Every verb is checked. If its morphological tag contains the symbol for negative verb, a new node is created with the lexical (`trlemma`) value ‘Neg’ and functor ‘RHEM’ (rhematizer, i.e. focus sensitive particle).

3.2.8 Other attribute assignments

Based on the morphological tag inherited from the analytical level of description, the values of the following morphological grammatemes are assigned: `gender` , `number`, `tense`, `degcmp` (degree of comparison), `aspect`.

The sentence modality is captured in the `sentmod` attribute of the head node of each clause. We assign the sentence modality of the head word of a simple sentence, of the main clause of a complex sentence and of all coordinated clauses in compound sentences. The sentence modality attribute value is determined by the final punctuation mark of the whole sentence and by the verb modality of the main verbs of the sentence clauses. The rules are described by Table 3.

Suppose we have a sentence composed of coordinated clauses X_i: X_1, X_2, ..., and X_n.

position in clause Xi	**final interp.**	**verb modali-ty**	**sentence modality of Xn**	**other conditions**	**verb modality assigned**
Xn (verb in the last or in the only clause)	**?**	-	-	-	INTER
	!	-	-	-	IMPER
	.	-	-	-	ENUNC
X1, ..., Xn-1 For n>1	-	-	INTER	-	INTER
	-	IND	-	-	ENUNC
	-	IMP	-	-	IMPER
	-	CDN	-	X_i contains ‘kéž’ (E:’let’)	DESID
	-	CDN	-	otherwise	ENUNC

Table 3. Sentence modality assignment

As for functors, their value is resolved automatically in the following three cases. Value ACT (actor/bearer, underlying subject) is assigned to every subject of an active verb. If there is a single object depending on an active verb, its node is assigned functor PAT (patient, objective). The head verbs of the sentences are assigned the functor PRED (predicate).

Example:

(i) **Sestra**.Sb **spatřila**.A **souseda**.Obj.
ACT PAT
E: *sister spotted the neighbour*

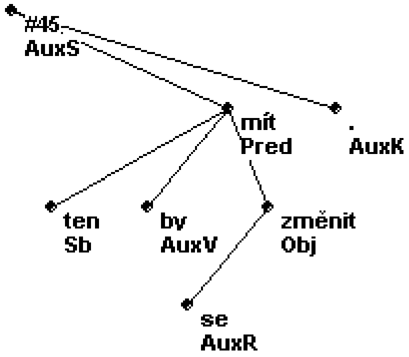
3.2.9 „Default“ values

Unresolved syntactic and morphological grammatemes are assigned their default value by the procedure. By the default value we understand ‘NIL’ value for attributes that cannot be assigned any value for the given node (e.g. case for verbal nodes), or it is chosen to express the uncertainty for the annotators (e.g. value "???" for unresolved `func` attribute).

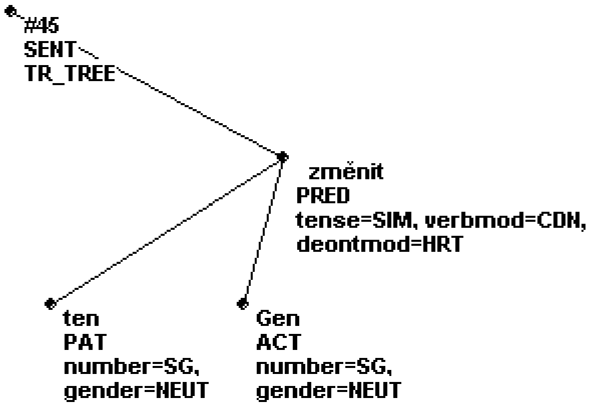
3.3 Example of input and output

Sentence: *To by se mělo změnit.*
That should (itself) change.
Meaning: *That should be changed.*

ATS:



TGTS:



References

[1] Hajič J. (1998) Building a syntactically annotated corpus: The Prague Dependency Treebank. In: *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová* (ed. by E. Hajičová) (pp. 106-132). Prague: Karolinum.

[2] Hajičová E. (1993) *Issues of sentence structure and discourse patterns.* Charles University.

[3] Hajičová E., B. H. Partee and P. Sgall (1998) *Topic-Focus Articulation, Tripartite Structures, and Semantic Content*. Dordrecht:Kluwer.

[4] Sgall P., E. Hajičová and J. Panevová (1986) *The meaning of the sentence in its semantic and pragmatic aspects,* ed. by J. L. Mey. Dordrecht:Reidel - Prague:Academia.

Considering Automatic Aids to Corpus Annotation

David Day and Benjamin Wellner
MITRE Corporation
Mail Stop K329
202 Burlington Road
Bedford, MA 01730, USA
day@mitre.org, wellner@mitre.org

Abstract

In this paper we view mixed-initiative corpus annotation from the perspective of knowledge engineering, and discuss some of the opportunities, challenges and dangers that are presented by using mixed-initiative annotation tools. We begin this discussion by describing an existing mixed-initiative annotation tool for open-ended phrase-level annotation, the Alembic Workbench. We discuss how this tool currently operates, the nature of its skill acquisition component, and our plans to extend it in a number of ways, including incorporating an active learning capability. Having set the stage with a concrete example, we identify a number of opportunities and challenges that are presented by the mixed-initiative approach to corpus annotation, including the benefits that might accrue when supporting "layered" annotation environments, the adoption of intensional/procedural annotation paradigms, the inclusion of lexical resource construction interleaved with corpus annotation, and other topics.

1 Introduction

One way of viewing corpus annotation is as a form of "knowledge engineering," where the *annotator* intends to enable a machine to reproduce the behavior being performed. A motivation for adopting such a view is that there is a practical interest in having machines be able to automatically perform some types of annotation. For example, "named entity" tagging, the ability to identify proper names that refer to entities of a particular restricted set of semantic classes (e.g., person, location, organization) was initially developed merely as a means to measure the contribution of this stage of linguistic analysis to a set of more complex domain-specific information extraction tasks. In recent years this capability has been shown to be valuable as a constituent to quite different information processing tasks, including topic detection and tracking, information retrieval, and others.

Another motivation for automating the annotation process is simply to increase the productivity of the corpus annotation process itself. Even if the ultimate goal of a particular annotation process is to build a static repository of annotated data to support fundamental linguistic research and analysis, there is a great benefit in producing as much of a given type of annotation as possible within restricted schedules and budgets. In general, the greater the size of the corpus, the more informed and statistically well-founded are the conclusions that can be drawn. From the point of view of knowledge engineering, most forms of corpus annotation involve a model of "learning by example," where some number of positive examples are meant to drive the skill acquisition component. In practice this skill acquisition is often carried out by a mix of human engineering (e.g., programming), machine-aided analysis, and machine learning techniques when possible. In this paper we want to expand on this skill acquisition model in a number of ways:

- Argue how these techniques can and should be applied across the full range of linguistic annotation tasks.
- Expand the notion of "mixed initiative" (or "incremental bootstrapping") annotation to incorporate not just learning by example, but other methods that increase the

expressive power of the "annotator" to influence skill acquisition.

- Encourage the use of "earlier" language processing stages in the annotation of later stages.
- Focussed corpus selection and annotation through "active learning" (or "sample selection").
- Common annotation frameworks and tools can help to increase these bootstrapping capabilities.

2 Mixed initiative corpus development

The notion of using partially machine-annotated data to "bootstrap" the human annotation process dates back at least to Brill's Ph.D. thesis (Brill, 1993), and probably earlier. The bootrapping procedure operates on the observation that there are many data points in some annotation tasks that are quite easily performed computationally. Even relatively poor performing procedures can prove effective for increasing productivity if there is a sufficiently large amount of data that is annotated correctly *and* if the labor required to fix the remaining bootstrapping errors is relatively small compared to the baseline manual tagging effort. In such a situation the bootstrapping procedure will have increased the effective productivity of the human annotator by the degree of the bootstrapping procedure's accuracy. For large corpus collections, this can represent a sizable savings in human labor.

The bootstrapping procedure can take many forms, and it can be arrived at in many ways, either through annotator-derived heuristics, systematic analysis of the corpus annotated so far, or through more automatic means utilizing machine/statistical learning techniques. We use the term *mixed initiative* annotation to refer to an environment in which (a) the bootstrapping procedure is derived automatically and (b) it can be invoked at arbitrary points during the course of annotation. (The alternative term "incremental bootstrapping" has also been suggested.) Subsequent invocations of the bootstrapping procedure can perform better than earlier invocations as a function of new evidence—usually in the form of a larger annotated corpus, since more examples are available to drive the bootstrapping procedure. Since first proposed, a number of tools have been built that provide mixed-initiative environments for a variety of annotation tasks.(Brants et al., 1997; Bennett et al., 1997)

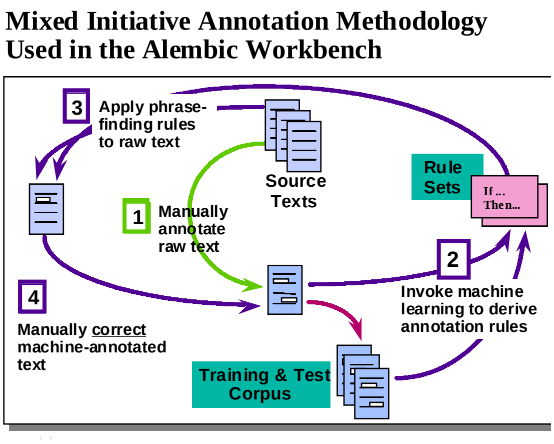

Mixed Initiative Annotation Methodology
Used in the Alembic Workbench
3
Apply phrase-finding rules to raw text
Source Texts
Rule Sets
If ...
Then...
1
Manually annotate raw text
2
Invoke machine learning to derive annotation rules
4
Manually correct machine-annotated text
Training & Test Corpus

Figure 1: The mixed-initiative spiral model of corpus annotation.

3 Mixed-initiative annotation in Alembic Workbench

We have developed a mixed-initiative annotation tool for building phrase-level annotated corpora; we will describe it briefly here in order to place in context our more recent work as well as to ground our subsequent discussion of general issues regarding the opportunities, challenges and dangers presented by mixed-initiative approaches to enhancing annotation productivity. This tool, the Alembic Workbench (Day et al., 1997) (or simply Workbench in this article), induces a finite state transducer in the form of a sequence of transformation rules, which can then be used to bootstrap (annotate) similar textual data. The transformation-based learner (TBL) and the transformational rule sequence interpreter are both provided by the Alembic multi-lingual natural language processing system (Aberdeen et al., 1995; Vilain and Day, 1996). A graphical depiction of the mixed-initiative methodology adopted by the Workbench is shown in Figure 1.

The class of annotations susceptible to mixed-initiative annotation in the Workbench might be best described as "phrasal"—any contiguous character sequence or multi-word sequence may be annotated by the user and associated with some "tag." The Workbench has been extended for a number of special purpose and general purpose annotation tasks, such as MUC6-style co-reference tagging, and general "relational" annotations. Neither of these tagging enhancements have yet been closely integrated with our machine-learning techniques—even though we are presently working on machine-learning approaches to both of these problems outside the specific scope of mixed-initiative annotation. These tags may be defined using the SGML/XML mechanisms for specifying generic identifiers of annotation elements and their associated attribute/value pairs. While this notational scheme allows for the expression of complex relationships among particular tags, the machine learning component currently used to support mixed-initiative annotation in the Workbench ignores these subtleties and reduces all SGML/XML elements with distinct structures into what are essentially unique, un-interpreted symbols.

The transformation-based rule sequence learning technique used in Alembic has proved very effective for deriving accurate annotation performance on the basis of exceedingly impoverished amounts of training data. We have observed F-measures in the range of 55-75 within 15-20 minutes of beginning a new annotation task in both English and Spanish texts, and lower but still helpful values in the same amount of annotation of Chinese and Japanese texts. (See section 3.1 for more background on the phrase-rule learning behavior of Alembic.) This is important in producing enhanced productivity through bootstrapping an automatic tagging procedure. The earlier the machine can provide pre-annotated data at a reasonable level of performance, the faster the combined activities of human and machine can build up a large corpus. In addition, the precision and recall of the automatically derived rule sequences increase as the size of the available training corpus increases. The shape of this learning curve is invariably asymptotic. The asymptote peaks at different levels (on blind test data) for different tasks and/or for different parameters of the learning environment, for reasons that are important, but not always easy to determine. The two main reasons appear to be the size and typicality of the corpus, and the representational power of the rule patterns available to the rule learner.

In building our current transformation-based learning (TBL) system we have made a number of design decisions that have had an impact on training speed. These decision have to do with the need to actually apply newly induced rules to the training data and subsequently recompute the corpus-wide statistics that drive the next rule induction cycle. While the system is able to achieve very fast learning times (5-30 minutes) on small to moderate amounts of training data (5,000 to 75,000 exemplars), much longer training times result when the training corpus reaches hundreds of thousands or even millions of exemplars.

Of course, there is no need to iterate the learning algorithm over all the training data after each document. Indeed, it seems reasonable and practical to invoke the learning algorithm less and less frequently as the size of the corpus increases and the performance of the automatically derived rule sequences differ less and less from each other with each incremental addition of a human-annotated file. In order to avoid having to restart the learning algorithm from an initial null state after each mixed-initiative invocation of the learning algorithm, it is straightforward within a TBL model to begin learning *on top of* of an existing level of competence (a previously derived rule sequence). In this case the training corpus might consist of only those few files that have been annotated since the last learning procedure was called, and the newly derived rules are concatenated onto the end of the existing rule sequence. However, eventually it is desirable to start fresh, since it becomes more and more likely that new opportunities for generalization can be found in a larger training set, leading to increases in the ability of the new rule sequence to apply successfully to unseen data.

Nonetheless, we are also interested addressing the problem of learning performance directly. One approach we intend to pursue is the incorporation of MITRE's HMM-based "Phrag" (Palmer et al., 1999) phrase-parsing learner within the Workbench's mixed-initiative repertoire, which we imagine could be increasingly relied upon as the size of the training set reaches very large proportions.

Currently the default "granularity" of mixed-initiative annotation within the Workbench is that of a document or file. As long as a single file is fully annotated, it can be used as the basis of phrase-rule learning, either alone or in combination with a corpus consisting of other annotated documents/files. Of course, documents can be arbitrarily reduced to smaller chunks if there is a strong need for this. Ideally one would like the granularity used in mixed-initiative annotation (1) to be identified and adopted directly by the system itself, rather than relying on the annotator to make such decisions; and (2) to be a function of the annotation task being performed. For example, phrase tagging (and many other annotation tasks such as sentence parsing) could be segmented at the sentence level. Updating the existing pre-annotation procedure could be invoked based on the amount of performance improvement achieved in the previous two invocations, as well as other heuristics that designers might identify. Other annotation tasks, such as co-reference annotation, discourse structure and entity and relation extraction, etc., might *require* segmentation at the document level.

3.1 Why Alembic phrase rule learning appears to work

In the past few years we have often been surprised at the ability of Alembic's phrase-rule learning apparatus to create quite reasonable tagging performance with only meager amounts of data annotated to the user's specifications. We have often had cause to wonder: Was our learning algorithm and associated Alembic infrastructure really so good? How did it squeeze out such good performance (e.g., around 70-75 F-measure) on such a paltry example base as 1,500 words of annotated Spanish newswire text? We would like to perform a detailed analysis, but our informal conclusions are already leading us to establish new priorities in our attempts to build rapidly portable natural language processing capabilities. We devote a subsection to each of these conclusions below.

3.1.1 The "right" level of analysis

Like many other systems designed in the course of the last five years, Alembic has been built using a number of important natural language processing components, each of which had become newly available in the previous years. These components include tokenization (word segmentation), sentence tagging, and morphological analysis (part-of-speech tagging). Empirically it has become clear that many useful types of general-purpose and specialized phrase tagging tasks (from named entity tagging to sentence chunking) can find all of the information they need from this mix of information made available by pre-processing. In other words, this is due to a successful application of the "divide and conquer" principle adopted by the computational linguistics community as a whole over the past ten years.

3.1.2 Locality of influence

In a similar vein, these same tagging tasks (perhaps best exemplified by "named entity tagging") have adopted a decision environment in which a fairly strict locality of influence is respected, and this locality has been sufficient for addressing the phrasal phenomena of interest. Not only has this seemed to be true for the rule schemata used in rule-based tagging systems, but also for the modeling techniques adopted in Hidden-Markov Model approaches to phrase tagging as well.

3.1.3 The "right" lexical resources and built-in predicates

We believe that the most specific reason that the Alembic phrase rule learner has managed to perform so well with very limited amounts of training data has been the considerable lexical resources that we have made available to the learner. It so happens that when Alembic has exhibited these surprisingly good learning behaviors it is often the case that the resulting rules include a liberal mixture of references to one or another of the special-purpose word lists that we have developed in the course of manually building various natural language processing capabilities. (Other frequently occurring rule patterns exhibited in successful rule sequences are those making use of the part-of-speech of lexical items. In CJK languages particular character prefixes and suffixes are also highly represented.)

These word lists are derived in a wide variety of ways: names extracted from the US Census; hand-coded lists expanded from core words easily predicted to be contextually important markers; expansions of words using thesauri, dictionaries or similar resources; words found from an analysis of the internal and external contexts of annotated phrases in manually tagged training data (supervised context analysis); and sometimes words found in these same contexts but from large collections of automatically tagged data (unsupervised context analysis). Regardless of the particulars of how they are derived, these resources allow for a boost in the generality of rules learned from a small corpus. While the learner might happen to pick references to these word lists for purely local (and perhaps almost arbitrary) reasons in the context of some very small annotated corpus, this serendipity will lead to many more correct applications when different word choices are encountered in previously unseen data.

Our current presumption is that replicating efficient mixed-initiative successes for other tasks and in other arenas of language processing will rely heavily on providing similar advantages as those identified above. For example, in order to support the rapid mixed-initiative annotation of certain types of relation/event data (e.g., the "template relation" and "scenario template" tasks of the various MUC evaluations (Def, 1995; Grishman and Sundheim, 1996)), one must make available to the learning component the same notion of "locality" as is warranted for such distinct phenomena. This kind of locality might be exemplified by an intermediate "SVO" (Subject/Verb/Object/modifier) representation of a given sentence, which could be derived in a variety of ways, either *via* treebank-style parses, or from dependency-like syntactic models such as "grammatical relations."). We are particularly interested in merging the mixed-initiative development of lexical resources with the mixed-initiative development of annotated corpora. (Previous work of others in this area includes (Riloff and Jones, 1999; Blum and Mitchell, 1998).) We anticipate that a host of unsupervised learning techniques will be especially useful in helping to quickly bootstrap the acquisition of useful word lists.

4 Active learning

One way of increasing the effective productivity of the human annotator while holding the capabilities of the skill acquisition component constant is by increasing the utility of the annotated data being supplied to the skill acquisition component. In the event that bootstrapping is being performed manually through heuristic insights, the annotator may try to tune the corpus sampling mechanism to favor sentences, paragraphs or documents that would seem to provide the greatest opportunity for instructing and testing the emerging automated annotation component. It is also possible to perform this sample selection of raw data through automatic means. This interplay between learner and example selection is sometimes referred to as "active learning" (or sample selection).(Lewis and Catlett, 1994)

Engelson and Dagan (Engelson and Dagan, 1996) demonstrated an automatic method for selecting part-of-speech training sentences using a votes from a set of automated annotation "experts." This and other work prompted us to look at how such techniques could be incorporated into the Workbench's mixed-initiative model. Alembic's phrase-rule learner contains a number of parameters that are well suited to construct such a family of experts.

The basic insight of active learning is that not all training data are equally informative, and that the "confidence" of the induced decision system in classifying (tagging) some particular exemplar is inversely proportional to the likely utility of that exemplar, were it to be correctly classified. If a particular annotation decision is made very confidently, it is likely due to the fact that many exemplars have informed the decision rule, and so increased the associated level of confidence. But how is "confidence" expressed in transformational rule sequences? In most cases, there is no analog to confidence in transformation rule sequences. However, if one can build a mixture of experts, then one analog to confidence in such systems is the number of experts that voted for the same tagging

1. Induce N different decision criteria by using varying parameter values.
2. Apply N decision criteria to unseen data.
3. Select for manual annotation those sentences for which there are sufficiently divergent classifications.
4. Annotate manually (with or without pretagging).

Figure 2: Active learning algorithm used in Alembic Workbench experiments

decision—independent of the nature of the decision mechanisms used in the constituent decision systems. The basic active learning algorithm used in our recent experiments with the Workbench is presented in Figure 2.

The Alembic transformation-based rule learning algorithm selects a rule at each epoch of the learning algorithm. We have experimented with a number of evaluation criteria for this step of the process: "yield minus sacrifice" (count the number of new, correct annotations created by applying a rule, then subtracting from this value the number of incorrect annotations created by applying this same rule); "log likelihood;" and "F-measure" (harmonic mean of the recall and precision measures for this rule), parameterized by *beta*, which indicates the relative weight given to the recall measure compared to the precision measure. We eventually adopted the F-measure approach, not only because it tended to give us the best empirical results on the problems we are addressing at that time, but also because it provided us the opportunity to transparently weight the performance more towards recall or more towards precision, which can be an important practical difference in various real world application contexts.

Varying the decision criterion by varying the *beta* value of the objective function allows us to easily define sets of experts from which "confidence" measures can be induced through their level of agreement. Indeed, the F-measure metric alone offers the opportunity for deriving a family of decision experts simply by modifying the single *beta* parameter. We would also like to use the Phrag HMM-based tagger on the same data to create an expert with a quite different bias. We are in the early stages of experimenting with this form of active learning, selecting sentences and/or documents on the basis of the degree to which multiple separately derived rule sequence "experts" agree on annotation assignments. These early results are encouraging. From an initial training set of 442 sentences (containing 705 target phrases), a subsequent unannotated corpus of 1,462 words was used as the universe of possible sentences for subsequent manual annotation. Approximately 10% were down-selected based on two different criteria: random selection or using low confidence measures as derived from voting as described above. Following the manual annotation of these two incremental additions to the training set, we observed that the performance of separately trained automatic taggers differed on test data by about 5% .[1]

5 Discussion

Corpus annotation has implications not just for providing productivity enhancements for linguists (computational and otherwise), but also as a model for how useful information extraction systems (an important class of intelligent agents) can be derived through a largely example-based knowledge engineering/acquisition process. With both of these contexts in mind, it is useful to reflect on some of the outstanding opportunities in mixed-initiative annotation, as well as the difficulties and dangers that accompany them.

5.1 Layered annotations for multi-staged mixed-initiative corpus development

Some annotation tasks depend so strongly on "earlier" annotations that it will become important to build annotation environments in which these earlier annotation "layers" are made apparent to the annotator. For example, when annotating *grammatical relations* (Ferro et al., 1999; Ferro, 1998), the job of the annotator is to establish various pre-specified types of relationships among sentence "chunks," where these chunks consist of simple phrases such as "noun group," "verb group," "preposition group," and

[1] We are in the midst of our explorations of this task; we hope to be able to report the results of more robust experiments soon.

the like. Thus, instead of being presented with a text in a standard Workbench textual display (and being able to draw relationships between arbitrary pairs of words), it is important that these sub-groupings are *already* visually apparent and made to control the interface so that asserting only group-level relationships is possible.[2]

Other opportunities for such interdependence of annotation tasks can be seen when annotating discourse-level relations and events (e.g., MUC-style "template relations" and "scenario templates"). While particular relationships may be asserted in a variety of ways, the ability to view and operate directly on, for example, an "SVO" (subject-verb-object-modifier) representation of a set of sentences might enhance not only the productivity of the annotator, but also build in important links across processing levels that are important to one's method of attacking a given computational linguistics problem. This ability to build upon the layers of annotation derived previously will become an increasingly important technique for building mixed-initiative annotation tools. It could prove especially fruitful in the support for a richer language by which the human annotator can directly influence the mixed-initiative process, as discussed in the next section.

5.2 From extensional to intensional annotation methods

We remarked earlier about how the bootstrapping of annotation can incorporate not just automatically derived annotation heuristics but also those derived from the human annotator, implemented usually as computer programs or simply regular expression macros. This has been a method frequently relied on within the computational linguistics community, since the skills for deriving the heuristics and implementing them as procedures are readily available. One of the open problems of mixed-initiative annotation environments is to provide some kind of support for more direct human intervention in the bootstrapping process other than simply adding yet another example. Of course, there a wide variety of pattern languages and annotation representations from which those inclined to write pre-annotation heuristics can choose. But are there ways in which the results of such heuristic annotation methods can be viewed and combined with example based annotations without creating confusion?

For example, if someone composes a rule and it applies to one hundred instances within a corpus, the annotator might like to view the resulting sentences directly—perhaps within a keyword-in-context type viewer—that is *also* integrated directly with the extensional annotation environment. This way exceptions to this rule can be noticed and modified easily and directly by the annotator. If this were truly a mixed-initiative environment, then such a system might on the next cycle derive a rule which *starts* with the human-authored heuristic, but derives rules (or some other representation) for capturing the exceptions identified extensionally by the annotator.

Interestingly, there has recently been a very careful empirical study (Brill and Ngai, 1999; Ngai and Yarowsky, 2000) exploring the advantages and disadvantages of extensional and intensional mixed-initiative methods for annotating a corpus. This study was carried out by Grace Ngai and David Yarowsky at Johns Hopkins University, and compared the abilities of relatively sophisticated pattern rule authors against machine learning methods for deriving tagging rules in a mixed-initiative annotation environment. The results indicate that rule-writing, while intuitively powerful, may prove difficult for supporting a mixed-initiative approach to corpus annotation. This is a provocative study, seeming at variance with our intuitions as computational linguists. The annotation community should explore these issues and discuss them fully.

5.3 The real world of task definition and collaborative development

In our own case studies and in our research focussed on mixed-initiative annotation we have often concentrated on well-defined annotation tasks and how they can most quickly be automated. In the real world, however, we know quite well from first-hand experience that the annotation process is a very long and tortuous

[2] Such an annotation tool has been developed specifically for the grammatical relations annotation task being performed internally at MITRE.

road, where many of the initial steps are concerned less with getting large amounts of annotated data quickly, but rather with exploring the very definition of the task at hand. As many of the contributors to formal language processing evaluations will tell you, much of the difficulty in starting up a new tagging task is due to the social and linguistic barriers to easy categorization. So how do the techniques we have described support and/or improve such task definition endeavors?

At the heart of any collaborative annotation effort is the detailed analysis and associated discussion of different interpretations of the linguistic phenomena, which is most often captured and brought to light through inter-annotator annotation analysis. At first this analysis is largely qualitative, and depends on *detecting* the anomalies in order to promote their discussion. Recently there has been a study of this collaborative behavior, and an associated automated method was developed that was modeled on it (Wiebe et al., 1999). Subsequently the emphasis moves towards quantitative inter-annotator analysis and the categorization of those differences. In both of these phases techniques that can boost the number and kinds of linguistic artifacts that have been annotated by one person or another can only help in the process of annotation understanding and inter-annotator reconciliation. Of course, it cannot sidestep the necessity of discussion and reflection that is necessary to come to terms with the motivations and other issues relevant to a new annotation task.

Nonetheless, there are clearly opportunities and challenges for mixed-initiative techniques that respect the collaborative nature of the annotation process. One area of interest is in building new automatic annotators by combining the existing annotation capabilities derived from separate human annotators interacting with mixed-initiative systems. For example, one could imagine new collaborative tasks could be defined through the application and analysis of distinct skills (tagging procedures, rule sequences, etc.) derived independently. This same ability may be appropriate for trying to identify and adapt to the inevitable “concept shift” that occurs with computational artifacts put to use on a daily basis.

5.4 The Tension between Naturally Occurring Phenomena and Focussed Inquiry

There is a potential danger that attends any technique that introduces labor saving methods, and mixed-initiative annotation is no exception. One of the most important problems is predicted to lie in the area of *recall*. As the automated pre-annotation process increases its capabilities, there will be a psychological tendency of human annotators to trust its guesses. And while precision errors will be fairly easy to spot (since the machine will display some text and assign a fallacious tag to it), *recall* errors—errors of omission—cannot be highlighted in principle, and so requires the human annotator to be forever vigilant and to notice “the tag that wasn’t.” This problem is perhaps accentuated even more with the adoption of active learning techniques. It is not known to what extent the introduction of active learning might introduce a vicious cycle of ignorance, whereby recall errors are never corrected due to tacit agreement (aligned errors) from all of the constituent decision components.

6 Conclusions

There are still opportunities for building, refining and applying mixed-initiative corpus annotation tools and environments. In this paper we have identified some of these opportunities, the challenges they pose and their potential for unintentional side effects. We grounded this discussion with a description of the Alembic Workbench tool, describing its current capabilities and the direction of our research to expand them. Successful mechanisms for quickly deriving machine-aided corpus annotation systems will have an important impact on the corpus linguistics research community. It will also lead eventually to portable, trainable language processing systems for use by non-specialists to perform customized information discovery and extraction from the glut of information available today.

References

John Aberdeen, John Burger, David Day, Lynette Hirschman, Patricia Robinson, and Marc Vilain. 1995. Description of the alembic system used for MUC-6. In *Proceedings of* *the Sixth Message Understanding Conference (MUC-6)*, pages 141–155, Columbia, Maryland, November.

Scott W. Bennett, Chinatsu Aone, and Craig Lovell. 1997. Learning to tag multilingual texts through observation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, Providence, RI, USA.

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *In Proceedings of the 11th Annual Conference on Computational Learning Theory. ACM.*

Thorsten Brants, Wojciech Skut, and Brigitte Krenn. 1997. Tagging grammatical functions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, Providence, RI, USA.

Eric Brill and Grace Ngai. 1999. Man vs machine: A case study in base noun phrase learning. In *Proceedings of Association of Computational Linguistics*. Association of Computational Linguistics.

Eric Brill. 1993. *A Corpus-Based Approach to Language Learning*. Ph.D. thesis, University of Pennsylvania, Philadelphia, Penn.

David Day, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson, and Marc Vilain. 1997. Mixed-initiative development of language processing systems. In *Fifth Conference on Applied Natural Language Processing*, Washington, D.C., March. Association for Computational Linguistics.

Defense Advanced Research Projects Agency. 1995. *Sixth Message Understanding Conference (MUC-6)*, Columbia, Maryland, November. Morgan Kaufmann.

Sean P. Engelson and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from corpora. *Computation and Linguistic E-Print Service*, cmp-lg/9606030, June.

L. Ferro, M. Vilain, and A. Yeh. 1999. Learning transformation rules to find grammatical relations. In *Computational natural language learning (CoNLL-99)*, pages 43–52. EACL'99 workshop, `cs.CL/9906015`.

L. Ferro. 1998. Guidelines for annotating grammatical relations. Unpublished annotation guidelines.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference: A brief history. In *International Conference on Computational Linguistics*, Copenhagen, Denmark, August. The International Committee on Computational Linguistics.

David Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning: Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156, San Francisco, CA. Morgan Kaufmann.

Grace Ngai and David Yarowsky. 2000. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *Proccedings of The 38th Annual Meeting*. Association for Computational Linguistics.

David D. Palmer, John D. Burger, and Mari Ostendorf. 1999. Information extraction from broadcast news speech data. In *Proceedings of the 1999 DARPA Broadcast News Workshop (Hub-4)*, February.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Nth National Coreference on Artificial Intelligence*. American Association for Artificial Intelligence.

Marc Vilain and David Day. 1996. Finite-state parsing by rule sequences. In *In International Conference on Computational Linguistics*, Copenhagen, Denmark, August. The International Committee on Computational Linguistics.

Janyce Wiebe, Rebecca Bruce, and Thomas O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics*, pages 246–253. Association of Computational Linguistics.