How one might Automatically Identify and Adapt to a Sublanguage: an initial exploration

Jonathan Slocum

Linguistics Research Center
University of Texas

Working Paper LRC-84-1
January, 1984

This paper was presented at the Sublanguage Workshop held at the Courant Institute of Mathematical Sciences, New York University, on January 19-20, 1984, sponsored in part by the National Science Foundation.

How one might Automatically Identify and Adapt to a Sublanguage: an initial exploration

Jonathan Slocum

Siemens Comm. Sys., Inc.
and The University of Texas

Introduction

This paper presents the results of the first study of "sublanguages" carried out at the Linguistics Research Center of the University of Texas, as part of the Machine Translation project. Our goal is the improvement of both the efficiency and quality of automated grammatical analysis of texts; we believe that these two issues (speed and quality) are closely related, in ways that will be explained below. Our approach here is to discover ways in which texts within a single sublanguage resemble each other, and how texts in different sublanguages will differ, then to propose a means for (semi)automatically identifying the sublanguage of a new text and optimizing a Natural Language Processing system for that text, so that overall performance may be improved.

The questions we most directly address, then, are these: Are there predictable characteristics of texts said to lie within a single sublanguage, and differences between texts said to be in different sublanguages (i.e., IS there such a phenomenon as sublanguage)? If so, how may these characteristics be described, and can the sublanguage of a text be automatically identified? If the sublanguage of a text can be identified, how does one construct a system that can quickly, automatically, and on-the-fly, optimize its performance for that text (actually, sublanguage)?

We will begin with a very brief overview of some of the relevant properties of the LRC Machine Translation system [Lehman et al., 1981], so that our means of gathering data, and our conclusions about how one might structure an adaptive system, will be apparent to the reader. Afterwards, we will describe the experimental setup in which we gathered our data, present and comment on the data, discuss the significance of our findings, and conclude with answers to the questions raised above, along with some commentary on the questions raised by the workshop organizers.

Overview of METAL

The LRC Machine Translation system is a collection of programs and data designed to automated the complete process of translating technical texts from one natural language into another. Programs include a relational database management system and several human interfaces to it (for maintaining dictionaries and grammar rules), a rule validation module (to check the syntactic integrity of all grammar rules and dictionary entries), a text-processing system (for automatically extracting from a formatted text the "sentences" to be translated, and reformatting the translation like the original), and METAL, which is the central translation engine.

METAL is composed of a set of dictionary and grammar rule definition modules, plus the linguistic rule interpreter which actually effects the analysis and translation of input "sentence" units [Slocum and Bennett, 1982]. For the purposes of this workshop, the main points of interest are the METAL parser, and the use of subject area ("provenience") tags in the dictionary entries.

The Parser

The METAL parser has evolved over the course of a number of years, as dictated by experience in attempting the analysis and translation of large volumes of (primarily German) text: approximately 1,000 pages in the past four years. Based on experiments begun at SRI International in 1978, and continued at the Linguistics Research Center [Slocum, 1980, 1981], we have used a simple, unadorned, all-paths Cocke-Kasami-Younger algorithm [Aho and Ullman, 1972], later augmented by top-down filtering [Pratt, 1973], and later a Left-Corner algorithm [Chester, 1980] similarly augmented (which resulted in its strongly resembling — if not actually being equivalent to -- the Earley algorithm [Earley, 1970]). Each step was taken only after large-scale experiments on real texts indicated that a significant efficiency gain would result.

Our latest steps have taken us out of the realm of purely all-paths parsers. Last Fall we began using a scheduler based on a static partial-ordering of the grammar rules: we have a stratified grammar, where all rules of a lower level are applied before any rules of a higher level, and where the parser ceases to apply any rules when the application of those on a given level has resulted in one or more analyses of the input sentence. Thus, we have a "some paths" parser, and our linguists tune the grammar rules (by means of leveling) so that -- if all goes well -- the "correct" parse is highly likely to be among the first interpretations found. The intent, obviously, is to avoid the production of many extraneous analyses. This technique resembles that of [Wotschke, 1975], though there are some important operational differences -- notably the lack of any "control graph" over the subgrammars. Our experience to date has amply demonstrated the effectiveness of this technique when coupled with other heuristic tools enabled by static rule ordering.

Most recently, we have supplanted the "static scheduler" (based strictly on rule level) with a "dynamic scheduler" (based on arbitrary heuristics, which currently include both static rule level and plausibility scores [Robinson, 1974, 1975]). In this way, we aim to have a system that can be tuned in accordance with the dictates of experience, so as to apply the most likely rules first and achieve analysis -- on the average -- much more efficiently. It is always possible that some sentences will cause the parser to thrash, but if overall average performance is improved we will have achieved our goal.

Provenience Tags

The METAL system has always assumed the existence of "tags" which indicate the subject-area(s) for which a given word or idiom is applicable. This is critical in our application, where the meaning (translation) of a term depends greatly on the technical area in which it is used. As it turns out, "technical area" is part of the semantics of the topic of this workshop: Sublanguage. Thus it should be no surprise when, later, we indicate how our tag scheme can be used to adapt the higher-level behavior of the METAL system to the analysis of specific sublanguages (METAL has been adapting its dictionary behavior for many years, on the basis of provenience tags).

The Experimental Setup

In order to gather data for this study, the METAL parser was instrumented to record the application of all grammar rules; this was made trivial by the fact that the evaluation of all rules is carried out by a single METAL subroutine. In addition, a special data analysis program was written to summarize and present the data thus gathered. Data points included the number of attempted applications of each individual grammar rule, the number of successes/failures that resulted (conditioned on subcategorization features, including semantic tests), and the number of times that the phrase [parse tree node] built by a successful rule actually appeared in a sentence-level parse tree.

We then searched our files for a set of four texts of approximately equal size, two each in (what were presumed to be) two different sublanguages; no attempt was made to arbitrarily equate the sizes of the texts since, for one thing, it is not obvious what criteria one might use without risking distortion of the results. The goal was to perform a factor analysis measuring the similarities of texts (supposedly) in the same sublanguage, and at the same time the differences between texts (supposedly) in different sublanguages. We found four such texts: two are [extracts from] operating/maintenance manuals for a Siemens digital telephone switching system, and the other two are essentially sales brochures from Computer Gesellschaft Konstanz of West Germany (a Siemens OEM subsidiary), describing and promoting certain Siemens computer systems that CGK deals in.

Even a brief glance at the texts reveals gross differences. The two manuals are primarily directive, while the two computer system sales brochures are descriptive in nature. All four German texts were analyzed and translated into English by METAL, without human intervention. Table 1 presents data about the sizes of the texts, their average "sentence" length, the portion actually parsed by METAL, the number of resulting interpretations, and for general information the total runtime (in "real" time) on a Symbolics LM-2 Lisp Machine. Needless to say, the four test runs were made under as close to the identical conditions as was possible; in particular, the identical system image was used, meaning that the grammar rules and dictionary entries, etc., were all the same.

It is clear from Table 1 that the sentences in the CGK texts average about three times the length of the sentences in the SIEmens texts. Since the texts were chosen for their approximately equal size in words, the number of sentences in the Siemens texts greatly exceeds the number of sentences in the CGK texts. (In point of fact, "sentence" must be taken figuratively, as technical texts frequently employ sentence units of simple phrases or even single words. The SIEmens manuals are especially notable for this.)

	SIEMAR	SIEAPR	CGKMAR	CGKAPR
#Words	1112	1157	1685	1243
#Pages @250 W/P	4.45	4.63	6.74	4.97
#Sentences	284	281	144	105
#Words/Sentence	3.9	4.1	11.7	11.8
#S's Parsed	236	258	105	80
(pct)	83%	92%	73%	76%
#Parses	461	481	703	534
#Parses/Parsed-S	1.95	1.86	6.70	6.68
Run Time	1h29m55s	1h18m33s	2h07m54s	2h20m50s
(sec)	5395	4713	7674	8450
#Sec/Word	4.85	4.07	4.55	6.80

Table 1
A Comparison of the 4 Texts
used for data acquisition

SIEMAR: a digital telephone system op/maint manual from Siemens, Mar. '83
SIEAPR: a digital telephone system op/maint manual from Siemens, Apr. '83

CGKMAR: a computer system description (sales mat'l) from CGK, Mar. '83
CGKAPR: a computer system description (sales mat'l) from CGK, Apr. '83

#Words: the number of words in the text (as sent from the sponsor for testing)
#Pages: the approximate size of the text, at 250 Words/Page

#Sentences: the exact number of sentences (or "sentence units") in the text
#Words/Sent: the average number of words per sentence

#Parsed: the number of sentences that resulted in one or more interpretations
(pct): the percentage of the "sentences" in the text that were parsed

#Parses: the total number of readings (interpretations) derived by the grammar
#Parses/S: the average number of readings/sentence, given that it was parsed

Run Time: the elapsed (real) time for the complete TRANSLATION of the text
#Sec/Word: the average number of (real time) seconds expended per word, for the complete translation run

The Experimental Data

Tables 2-5 summarize the data by grammatical category, for each of the four German texts. Each table presents a complete accounting of every grammar rule called (organized by Left-Hand-Side category), and the outcome of the attempt (in terms of the numbers of local successes [phrases built], failures [rules rejected for violating subcategorization conditions], and the number of phrases which eventually appeared in S-level interpretations [parse trees]). The latter number can exceed the number of phrases accepted due to sharing of nodes among multiple parse trees.

A brief review of the data reveals what one would expect: the CGK texts seem to be richer, with "more of everything" in the way of syntactic variety. For example, by looking in each successive table at the categories ADJ, ADV, NN [NouN], and VB -- which with other constituents build to the higher-level categories NP, PP, CLS [CLauSe, including main, RELative, and SUBordinate varieties] -- it is obvious that the CGK texts exhibit more syntactic phenomena than the Siemens texts. With a little closer study, it also becomes clear that the texts do fall into two categories -- i.e., the two CGK texts lie in one "sublanguage" [as defined by syntactic characteristics], while the two Siemens texts fall into another. Table 6 eases this comparison; alternate columns represent the absolute and rank-order (w.r.t. appearances in S readings) frequencies of occurrence of phrases in the various categories.

It also becomes clear, even at this superficial level, that the syntactic phenomena in the Siemens texts are NOT simply a subset of those in the CGK texts. Most obviously, there are constructs in the one that are entirely unrepresented in the other: parenthetical phrases of various kinds, and "ZU CLauses" (characteristic of the German equivalent of the English "in order to" construct). Thus, the language in the Siemens texts is not a "subset" of the language in the CGK texts. (Prescriptive inspection also reveals that the Siemens texts are not a "subset" of acceptable German, either, though the CGK texts appear to be so -- perhaps because the former were written by engineers, while the latter were presumably written by sales personnel.)

A deeper analysis of the full data set reveals some even more interesting findings. Table 7 is a breakdown of some selected individual grammar rules, from which it is possible to discern, not only what categories of phrases were built, but also HOW they were built (i.e., with what constituent structure). We have chosen clauses, nouns, and noun phrases for this illustration.

The rule CLS = (NP RCL) takes a Right-branching CLause (in German, a portion of a sentence with a finite verb/auxiliary at the front) and adds a complement NP (e.g., a subject or direct object) to it. This is the most frequently represented CLS rule in three of the four texts (clauses are usually finite, have subjects, and most have direct objects as well); it is much more prevalent in the CGK texts since so many "sentences" in the Siemens texts are just nouns or noun phrases. However, the rule that adds a PP complement is far more obviously common in the CGK texts than in the Siemens texts, because the longer CGK sentences have many more complements to add.

The rule CLS = (NFCL) appears more often in the Siemens texts (indeed, it appears only once in the two CGK texts combined) because this construct (in our grammar) is characteristic of an imperative -- much more likely to occur in an operating/maintenance manual.

The rule CLS = (CLS PNT REL) adds a relative clause (actually modifying one of the constituents of the CLS) which has been separated from its modificand. [A transformation in the body of this PS rule produces a phrase structure representing the proper association.] The CGK texts have many more relative clauses than do the Siemens texts.

the NN rules indicate that the relative order of noun types (stems, inflected forms, and acronyms) is the same in all texts -- but the relative number of acronyms in the Siemens texts is much higher than in the CGK texts. [Engineers love acronyms.]

The NP rules reveal a striking reversal of NP types: while the CGK texts display the usual pattern of German whereby nouns are very likely to have determiners (even when English would not), the Siemens texts indicate that engineers writing manuals prefer to drop the determiners (as they tend to drop most other things). Similarly, the CGK salesmen like to modify NP's with PP's, while the engineers do not; likewise, they are more likely to employ appositive (NP NP) constructs and conjoined NP's. Finally, the CGK texts display a far higher incidence of pronouns than do the Siemens texts. Thus, for example, pronominal resolution is likely to be a much more severe problem in the sales domain than in that of operating/maintenance manuals.

Cat	#Calls	#Succs	pct	#Fails	pct	#Apps	#C/#A
ADJ	2094	462	22	1632	77	176	11
ADJ-LCL	1695	699	41	996	58	25	67
ADV	1163	55	4	1108	95	29	40
AST	363	5	1	358	98	1	363
AST-LCL	542	71	13	471	86	25	21
CLS	4325	670	15	3655	84	190	22
CLS-REL	2484	435	17	2049	82	3	828
CLS-SUB	2666	30	1	2636	98	4	666
COMP	719	392	54	327	45	3	239
CON	8	3	37	5	62	0	99999
CONJ	97	0	0	97	100	0	99999
DEG	1	1	100	0	0	0	99999
DET	46	0	0	46	100	0	99999
HYPHCLS	24	0	0	24	100	0	99999
HYPHNP	12	0	0	12	100	0	99999
HYPHPP	4	0	0	4	100	0	99999
LCL	1336	1218	91	118	8	18	74
NFCL	674	574	85	100	14	369	1
NFPRED	610	178	29	432	70	152	4
NN	2125	663	31	1462	68	879	2
NO	1450	915	63	535	36	1057	1
NP	18830	2798	14	16032	85	1220	15
NPMOD	3	1	33	2	66	0	99999
NST	338	46	13	292	86	14	24
PARADJ	4	4	100	0	0	2	2
PARAV	5	5	100	0	0	0	99999
PARCLS	7	5	71	2	28	0	99999
PARNP	10	10	100	0	0	12	0
PP	672	216	32	456	67	207	3
PRED	586	227	38	359	61	45	13
PREP	3	0	0	3	100	0	99999
PRFX	353	104	29	249	70	6	58
PRN	167	88	52	79	47	13	12
PRT	3	0	0	3	100	0	99999
RCL	1780	882	49	898	50	159	11
S	831	509	61	322	38	461	1
VB	15585	528	3	15057	96	244	63
VBMOD	1	1	100	0	0	0	99999
ZUCL	721	156	21	565	78	12	60

Table 2
Summary of Rule Applications
for the text SIEMAR

Cat: the grammatical category (of a set of 1+ PS rules)
#Calls: the number of applications of rules attempted by the parser
#Succs: the number of rules successfully applied
#Fails: the number of rules rejected on sub-categorization grounds
#Apps: the number of appearances of the phrase-type in S readings
#C/#A: the ratio of #Calls to #Appearances (a measure of utility)
[99999 = "infinite"]

Cat	#Calls	#Succs	pct	#Fails	pct	#Apps	#C/#A
ADJ	2270	427	18	1843	81	184	12
ADJ-LCL	284	110	38	174	61	38	7
ADV	771	26	3	745	96	10	77
AST	274	3	1	271	98	4	68
AST-LCL	545	68	12	477	87	36	15
CLS	3786	1501	39	2285	60	212	17
CLS-REL	1653	1091	66	562	33	13	127
CLS-SUB	2217	8	0	2209	99	8	277
COMP	666	320	48	346	51	13	51
CON	12	7	58	5	41	0	99999
CONJ	65	2	3	63	96	0	99999
DEG	1	1	100	0	0	0	99999
DET	76	0	0	76	100	0	99999
HYPHCLS	18	0	0	18	100	0	99999
HYPHNP	7	0	0	7	100	0	99999
HYPHPP	3	0	0	3	100	0	99999
LCL	2313	1518	65	795	34	44	52
NFCL	845	548	64	297	35	279	3
NFPRED	647	202	31	445	68	173	3
NN	2512	696	27	1816	72	786	3
NO	1424	974	68	450	31	1084	1
NP	10616	2156	20	8460	79	1116	9
NPMOD	3	1	33	2	66	0	99999
NST	174	27	15	147	84	41	4
PARCLS	5	3	60	2	40	20	0
PARNP	12	12	100	0	0	14	99999
PP	879	345	39	534	60	199	4
PRED	614	244	39	370	60	79	7
PRFX	344	104	30	240	69	4	86
PRN	100	45	45	55	55	18	5
PRT	8	0	0	8	100	0	99999
RCL	948	570	60	378	39	229	4
S	832	504	60	328	39	481	1
VB	15251	538	3	14713	96	311	49
VBMOD	1	1	100	0	0	0	99999
ZUCL	596	63	10	533	89	40	14

Table 3
Summary of Rule Applications
for the text SIEAPR

Cat: the grammatical category (of a set of 1+ PS rules)
#Calls: the number of applications of rules attempted by the parser
#Succs: the number of rules successfully applied
#Fails: the number of rules rejected on sub-categorization grounds
#Apps: the number of appearances of the phrase-type in S readings
#C/#A: the ratio of #Calls to #Appearances (a measure of utility)
[99999 = "infinite"]

Cat	#Calls	#Succs	pct	#Fails	pct	#Apps	#C/#A
ADJ	2409	799	33	1610	66	843	2
ADJ-LCL	393	254	64	139	35	189	2
ADV	814	101	12	713	87	169	4
AST	529	24	4	505	95	21	25
AST-LCL	725	79	10	646	89	111	6
CLS	21570	5163	23	16407	76	1071	20
CLS-REL	568	467	82	101	17	185	3
CLS-SUB	3136	147	4	2989	95	246	12
COMP	1684	717	42	967	57	193	8
CON	16	2	12	14	87	36	0
CONJ	210	9	4	201	95	8	26
DEG	1	1	100	0	0	0	99999
DET	249	8	3	241	96	0	99999
HYPHCLS	28	0	0	28	100	0	99999
HYPHNP	27	0	0	27	100	0	99999
HYPHPP	1	0	0	1	100	0	99999
LCL	3057	2650	86	407	13	1280	2
NFCL	2246	1785	79	461	20	1072	2
NFPRED	928	271	29	657	70	514	1
NN	2533	668	26	1865	73	3080	0
NO	1644	910	55	734	44	3847	0
NP	11595	1827	15	9768	84	4535	2
NPMOD	13	3	23	10	76	0	99999
NST	440	38	8	402	91	14	31
PP	1253	445	35	808	64	1272	0
PRED	814	287	35	527	64	812	1
PREP	6	2	33	4	66	0	99999
PRFX	567	395	69	172	30	29	19
PRN	332	159	47	173	52	54	6
RCL	3022	1650	54	1372	45	1399	2
S	754	742	98	12	1	703	1
VB	10706	717	6	9989	93	1272	8
VBMOD	6	0	0	6	100	0	99999
ZUCL	765	37	4	728	95	0	99999

Table 4
Summary of Rule Applications
for the text CGKMAR

Cat: the grammatical category (of a set of 1+ PS rules)
#Calls: the number of applications of rules attempted by the parser
#Succs: the number of rules successfully applied
#Fails: the number of rules rejected on sub-categorization grounds
#Apps: the number of appearances of the phrase-type in S readings
#C/#A: the ratio of #Calls to #Appearances (a measure of utility)
[99999 = "infinite"]

Cat	#Calls	#Succs	pct	#Fails	pct	#Apps	#C/#A
ADJ	1303	268	20	1035	79	372	3
ADJ-LCL	791	331	41	460	58	124	6
ADV	364	42	11	322	88	89	4
AST	208	10	4	198	95	62	3
AST-LCL	573	91	15	482	84	106	5
CLS	31700	10155	32	21545	67	511	62
CLS-REL	1652	1330	80	322	19	91	18
CLS-SUB	5384	193	3	5191	96	306	17
COMP	1213	678	55	535	44	94	12
CON	4	0	0	4	100	0	99999
CONJ	151	1	0	150	99	0	99999
DEG	1	1	100	0	0	6	0
DET	64	3	4	61	95	0	99999
HYPHCLS	30	0	0	30	100	0	99999
HYPHNP	25	0	0	25	100	0	99999
HYPHPP	54	0	0	54	100	0	99999
LCL	5528	4134	74	1394	25	1107	4
NFCL	2391	1696	70	695	29	580	4
NFPRED	833	198	23	635	76	688	1
NN	1428	431	30	997	69	2111	0
NO	848	579	68	269	31	2831	0
NP	23888	1911	7	21977	92	3600	6
NPMOD	15	0	0	15	100	0	99999
NST	377	21	5	356	94	8	47
PP	1300	631	48	669	51	1019	1
PRED	967	305	31	662	68	574	1
PREP	8	0	0	8	100	0	99999
PRFX	845	691	81	154	18	67	12
PRN	234	118	50	116	49	16	14
PRT	3	0	0	3	100	0	99999
RCL	3237	1704	52	1533	47	1009	3
S	660	559	84	101	15	534	1
VB	15815	564	3	15251	96	1072	14
VBMOD	3	0	0	3	100	0	99999
ZUCL	1171	462	39	709	60	0	99999

Table 5
Summary of Rule Applications
for the text CGKAPR

Cat: the grammatical category (of a set of 1+ PS rules)
#Calls: the number of applications of rules attempted by the parser
#Succs: the number of rules successfully applied
#Fails: the number of rules rejected on sub-categorization grounds
#Apps: the number of appearances of the phrase-type in S readings
#C/#A: the ratio of #Calls to #Appearances (a measure of utility)
[99999 = "infinite"]

	SIEMAR		SIEAPR		CGKMAR		CGKAPR	
Category	#App	#App/#S-Int	#App	#App/#S-Int	#App	#App/#S-Int	#App	#App/#S-Int
S	461	1.0	481	1.0	703	1.0	534	1.0
ADJ	176	.3817787	184	.3825364	843	1.199147	372	.6966292
ADJ-LCL	25	.05422993	38	.07900208	189	.2688478	124	.2322097
ADV	29	.06290672	10	.02079002	169	.2403983	89	.1666667
AST	1	.002169197	4	.008316008	21	.02987198	62	.1161049
AST-LCL	25	.05422993	36	.07484407	111	.1578947	106	.1985019
CLS	190	.4121475	212	.4407484	1071	1.523471	511	.9569288
CLS-REL	3	.006507592	13	.02702703	185	.2631579	91	.170412
CLS-SUB	4	.00867679	8	.01663202	246	.3499289	306	.5730337
COMP	3	.006507592	13	.02702703	193	.2745377	94	.17603
CON	0		0		36	.0512091	0	
CONJ	0		0		8	.0113798	0	
DEG	0		0		0		6	.01123596
DET	0		0		0		0	
HYPHCLS	0		0		0		0	
HYPHNP	0		0		0		0	
HYPHPP	0		0		0		0	
LCL	18	.03904555	44	.09147609	1280	1.820768	1107	2.073034
NFCL	369	.8004338	279	.5800416	1072	1.524893	580	1.086142
NFPRED	152	.329718	173	.3596674	514	.7311522	688	1.28839
NN	879	1.906725	786	1.634096	3080	4.381223	2111	3.953184
NO	1057	2.292842	1084	2.253638	3847	5.472262	2831	5.301498
NP	1220	2.646421	1116	2.320166	4535	6.450925	3600	6.741573
NPMOD	0		0		0		0	
NST	14	.03036876	41	.08523909	14	.01991465	8	.01498127
PARADJ	2	.004338395	0		0		0	
PARAV	0		0		0		0	
PARCLS	0		20	.04158004	0		0	
PARNP	12	.02603037	14	.02910603	0		0	
PP	207	.4490239	199	.4137214	1272	1.809388	1019	1.90824
PRED	45	.09761388	79	.1642412	812	1.15505	574	1.074906
PREP	0		0		0		0	
PRFX	6	.01301518	4	.008316008	29	.04125178	67	.1254682
PRN	13	.02819957	18	.03742204	54	.07681366	16	.02996255
PRT	0		0		0		0	
RCL	159	.3449024	229	.4760915	1399	1.990043	1009	1.889513
VB	244	.5292842	311	.6465696	1272	1.809388	1072	2.007491
VBMOD	0		0		0		0	
ZUCL	12	.02603037	40	.08316008	0		0	

Table 6
Absolute and Relative # Appearances
of grammatical phrases
in S interpretations

#App: the total number of appearances of a phrase of the given category in all interpretations of the sentences actually parsed

#App/S: the average number of appearances of a phrase of the given category per sentence interpretation

Syntax Rule		Number (and Relative Frequency) of Appearances in Texts			
LHS	RHS	SIEMAR	SIEAPR	CGKMAR	CGKAPR
CLS	(NP RCL)	83(1)	43(1)	464(1)	133(2)
CLS	(PP RCL)	7(4)	23(3)	192(2)	157(1)
CLS	(RCL)	17(3)	35(2)	0(*)	36(3)
CLS	(NFCL)	59(2)	10(8)	0(*)	1(*)
CLS	(CLS PCT REL)	3(*)	11(7)	144(3)	31(5)
NN	(NST)	486(1)	528(1)	1982(1)	1648(1)
NN	(NST N-FLEX)	216(2)	164(2)	1082(2)	375(2)
NN	(ACRON)	176(3)	93(3)	15(3)	87(3)
NP	(NO)	510(1)	544(1)	969(2)	1043(2)
NP	(DET NO)	347(2)	234(2)	2039(1)	1245(1)
NP	(NP PP)	105(4)	115(3)	446(3)	340(3)
NP	(NP NP)	164(3)	111(4)	263(5)	335(4)
NP	(NP CONJ NP)	1(*)	10(*)	231(6)	158(5)
NP	(PRN)	16(5)	21(6)	284(4)	107(6)

Table 7
Breakdown of Selected PS Rules
appearing in final parse trees

CLS: [main] CLauSe
NFCL: Non-Finite CLause
RCL: Right-branching CLause
REL: RELative clause

NP: Noun Phrase
PP: Prepositional Phrase

NO: NOminal (e.g., a NouN plus modifying adjectives)

NN: NouN (built from a stem and an [optional] inflectional ending)

ACRON: ACRONym

CONJ: CONJunction
DET: DETerminer
NST: Noun STem
PCT: PunCTuation (e.g., a comma)
PRN: PRoNoun

N-FLEX: a noun inflectional ending

Discussion

Knowing about the existence of sublanguages is of little value unless one can take advantage of this knowledge in some fashion. Here we will discuss how one might detect the existence of a particular sublanguage, and consequently adjust parameters that optimize the system for that sublanguage.

Detection

There are two obvious, not necessarily mutually exclusive means whereby one could identify the sublanguage of a new text. First, the system can ask the user about the text -- probably through a menu-type of interface -- in terms that are easy to comprehend and reliably respond to (e.g., "Is this a manual?" and/or "What [technical] subject area does this text cover?"). Second, the system could scan the text, looking up the words in its dictionary and determining from relative frequencies of pre-stored subject-area tags what the most likely topic of the text is. (Walker and Amsler, at this workshop, discuss such a technique.) This might not suffice to identify the type of text (e.g., a manual), but then this is not yet known to be the case: one might, for example, compare the number of determiners with the number of nouns and/or consider the relative incidence of acronyms.

We recall that the METAL system has always employed subject-area tag coding in dictionary entries for translation purposes [and also for idiom analysis]. It would seem that we should be able to make use of these tags to automatically identify the subject-area of any text at hand, so long as it lies in one of the areas covered by the system dictionary. (If the text lies outside the METAL's lexical domain, the system cannot be used effectively in any case.) Thus, a completely automatic determination of the provenience area of any text in the areas covered by the dictionary seems feasible.

Deriving Advantage

Again, knowing what sublanguage is in use is not of itself valuable: one must be able to take beneficial action based on such knowledge. We recall the new METAL dynamic scheduler [invented for reasons entirely independent of the existence of sublanguage]. The grammar rules are manually stratified (assigned to one of a number of static "levels") by the LRC linguists. In its static form, the scheduler caused the parser to invoke all possible lower-level rules before any higher-level ones; in its new, dynamic form, the parser schedules rule application by a combination of static level and a plausibility factor (a "weight" attached to each phrase satisfying a rule constituent). That is, certain phrase readings are naturally preferred over others [we use the weights at the S level to select "the best" analysis for translation], and the dynamic scheduler attempts to alter the parser's activity by using these weights to bias the otherwise-static rule stratification.

The experimental data presented here indicate that there are significant differences in the syntactic rule sets, and consequently their optimal application order, vis-a-vis the particular sublanguage. Findings like these are supported by other workers in the field (e.g., [Kittredge and Lehrberger, 1982]). The METAL dynamic scheduler can easily be modified so that the rule selection strategy is biased by the identification of the sublanguage of the text at hand. We intend to perform this modification and carry out further experiments along this vein in the near future.

Conclusions

We have independently determined that "sublanguages" do indeed appear to exist (i.e., that there seem to be reliable and measurable differences), and furthermore that sublanguages can be described on syntactic grounds (among others). We have adduced two simple, inexpensive techniques for automatically identifying the sublanguage of a text. We have described how at least one NLP system (METAL) can be modified to take advantage of sublanguage identification [even more than it does already] using tools already present in the system. What remains to be seen -- and what we will address in future experiments -- is whether, and to what extent, the advantage gained will be significant. We have reason to believe that such modification will not only enhance the runtime performance of our Machine Translation system by reducing the number of grammar rules applied [currently the limiting performance factor in METAL], but will also improve the quality of its translations by further reducing the number of incorrect readings that compete for translation attention.

Regarding some of the questions raised by the conference organizers, we can make the following comments based on our experience (including the experiment reported herein). We are not aware of any sublanguages for which any NLP system is currently able to obtain "correct sentence analyses with high reliability." However, the data we present here (and our in-house examination of the translation results produced by these runs) indicate that the METAL system now appears to perform in the 80% range for documents like the Siemens texts used here; furthermore, based on our history of continual quality improvement [Slocum, 1983], we see no particular reason why a 90% accuracy figure for such manuals could not be attained with current technology. [In our particular situation, unfortunately, Siemens has recently directed us away from the telephone manuals toward other -- much more difficult -- types of material, such as the CGK sales brochures used here; thus, we ourselves do not expect to attain 90% reliability in the foreseeable future. Our conjecture must therefore be taken with the proper dose of salt.]

It is certainly the case that, for "ultimate understanding" in ANY domain, an NLP system will have to be augmented with a wide variety of powerful tools for syntactic, semantic, and pragmatic analysis. For "appropriate response" at the 90% level within some sublanguages, this may not be necessary. We certainly hope that such powerful tools are NOT necessary, since it is obvious that they will not exist for quite some time -- probably not in this century. For example, little if anything is known about "discourse structures" of any kind that can be used in an NLP system with even minimal reliability (50%?) in large-scale application. Indeed, exceptionally few NLP workers have made a serious attempt at large-scale application of the techniques they espouse.

As for representation, there seems to be no objective evidence whatever that one school of thought is necessarily superior to any other. No one has tried to come up with empirical evidence bearing on these arguments, and such questions as are raised about sufficiency in application are banished to the rarely trod ("uninteresting") realms of "implementation details." In such a climate, objective arguments are difficult to muster.

Accordingly, little information can be discovered "in an automatic or semi-automatic fashion for a new domain". But it would seem that the type of data we present here can be used to automatically tune a grammar for syntactic reliability. Whether this is ultimately beneficial has yet to be determined.

References

Aho, A. V., and J. D. Ullman. The Theory of Parsing, Translation, and Compiling, vol. 1. Prentice-Hall, Englewood Cliffs, N.J., 1972

Chester, D., "A Parsing Algorithm that Extends Phrases," AJCL 6 (2), April-June 1980, pp. 71-86.

Earley, J., "An Efficient Context-free Parsing Algorithm," CACM 13 (2), Feb. 1970, pp. 94-102.

Kittredge, R., and J. Lehrberger. Sublanguage: Studies of Language in Restricted Semantic Domains. de Gruyter, New York, 1982.

Lehmann, W. P., W. S. Bennett, J. Slocum, et al., "The METAL System," Final Technical Report RADC-TR-80-374, Rome Air Development Center, Griffiss AFB, New York, January 1981. Available as Report AO-97896, National Technical Information Service, U.S. Department of Commerce, Springfield, Va.

Pratt, V. R., "A Linguistics Oriented Programming Language," Proceedings of the Third International Joint Conference on Artificial Intelligence, Stanford University, California, 20-23 August 1973, pp. 372-381.

Robinson, J. J., "Performance Grammars," Proceedings of the IEEE Speech Symposium, Carnegie-Mellon University, Pittsburgh, 15-19 April 1974.

Robinson, J. J., "A Tuneable Performance Grammar," presented at the Thirteenth Annual Meeting of the ACL, Boston, 30 October - 1 November 1975.

Slocum, J., "An Experiment in Machine Translation," Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics, Philadelphia, 19-22 June 1980, pp. 163-167.

Slocum, J., "A Practical Comparison of Parsing Strategies for Machine Translation and Other Natural Language Processing Purposes," University Microfilms International, Ann Arbor, Mich., 1981.

Slocum, J., and W. S. Bennett, "The LRC Machine Translation System: An Application of State-of-the-Art Text and Natural Language Processing Techniques to the Translation of Technical Manuals," Working Paper LRC-82-1, July, 1982.

Slocum, J., "A Status Report on the LRC Machine Translation System," Proceedings of the ACL Conference on Applied Natural Language Processing, Santa Monica, California, 1-3 February 1983.

Wotschke, E.-M., "Ordered Grammars with Equivalence Classes: some Formal and Linguistic Aspects," Ph.D. dissertation, UCLA, 1975.