XI. SYNTACTIC ANALYSIS OF INCOMPLETE SENTENCES IN THE SMART SYSTEM

James Prowse

1. Introduction

One feature of the SMART information retrieval system[1] is the optional syntactic analysis of search request, documents, and document titles to provide additional information in making relevance judgments. This syntactic analysis is performed by the Harvard Multiple-path English Syntactic Analyzer, developed by Kuno and Oettinger. The analyzer consists of a package of three programs together with an English grammar on tape. Input to the first section is a specially prepared English text; output from the third and final section is a printable analysis tape giving all analyses compatible with the grammar for each input sentence. In SMART this package has been adopted virtually without change, the exceptions being alteration of tape assignments and a simple modification to terminate processing with the first analysis obtained for any input string. In the English analyzer, sentences are considered well-formed relative to the grammar only if they satisfy the usual criteria for completeness, that is if they contain both a subject and a predicate. Unfortunately this requirement renders impossible analysis of certain document titles, for example, "On Numerical Methods for High Speed Computation," or "A new FORTRAN Compiler for the IBM 360". As indicated by the examples, these ill-formed (relative to the analyzer) titles can have a noun phrase or a prepositional phrase structure; investigation has confirmed the hypothesis that these two

types, together with a complete sentence, comprise the forms taken by virtually all titles of scientific documents. This paper describes modifications to the SMART version of the English syntactic analyzer which permit analysis of such incomplete sentences.

2. A Brief Description of the Analyzer

The multiple-path syntactic analyzer has been extensively documented[2,3] and will not be described in detail here. It operates by the method of predictive syntactic analysis, using a pushdown store called the prediction pool. Analysis proceeds from left to right, a sentence prediction having been initially placed in the prediction pool. The input string consists of a series of sets of homographs, one set for each word in the sentence. The homographs determine membership in syntactic word classes, for example the homographs assigned to "autumn" are NOUS, singular noun, and NAD, noun adverb. To begin analysis, the sentence prediction in the pool is combined with the first homograph of the first word of the sentence; this argument pair is then looked up in the grammar table, obtained from the grammar tape. If the syntactic word class corresponding to this homograph can begin a sentence, one or more sets of new predictions, called subrules, will be found in the grammar table. The first of these sets replaces the sentence prediction in the prediction pool, and analysis continues to the next word position. On the other hand, if no entry is found in the grammar table, the second homograph of the first word is matched with the sentence prediction and a second grammar search is made. By means of a systematic loop-free sequence all homographs are tried

at each word position, and each set of new predictions obtained from the grammar by a particular argument pair is entered into the prediction pool at least once. The multiple-path technique ensures that any path which could reach the last word of the sentence will be tried; if no such path exists, analysis fails for this sentence. Each successful path corresponds to an acceptable syntactic structure relative to the grammar, but in the SMART version only the first of the many possible analyses for a single sentence is obtained. Analysis is simply terminated after one path is found which reaches the last word of the sentence.

An aspect of the analyzer important in this paper is its treatment of prepositional phrases. These structures are never predicted during analysis but instead are accepted as floating structures whenever they appear. For example, in the analysis of the sentence "Rand has written a new FORTRAN compiler for the IBM 360", a period prediction remains in the pool after "Rand has written a new FORTRAN compiler" has been accepted. The grammar contains a set of subrules whose argument pair is 'period, preposition', permitting the period prediction to complete analysis by accepting the prepositional phrase which terminates the sentence. This power of the period prediction to accept recursively the floating structures appended to a basic sentence becomes crucial when incomplete sentences are considered.

3. Methods for Solving the Acceptance Problem

There are two ways by which titles which are incomplete relative to a full sentence might be analyzed. The first is to modify the analyzer programs

internally so that failure to obtain any analysis signals a new attempt under different initial conditions. After failure of the first try with a sentence prediction initially in the pool, a second try could be made with the sentence prediction replaced by a noun phrase prediction. Analysis would then proceed as usual, and if failure resulted from this second pass, a third try would be made with the pool initialized by a prepositional phrase prediction. Only after a third failure would analysis be terminated as unsuccessful.

There are two serious drawbacks to this first method. The first has already been mentioned in Part 2: in order to accept a noun phrase followed by a prepositional phrase ("A New FORTRAN Compiler for the IBM 360") or a prepositional phrase followed by a prepositional phrase ("On Numerical Methods for High Speed Computation") not only must the noun phrase prediction or prepositional phrase prediction be loaded into the pool, but a period prediction must also be supplied. The second difficulty results from the separation of the analyzer package into three distinct programs. Program two, SYNTAX, performs the actual analysis while program three, EDIT, produces an edited version of the raw binary analysis output supplied by SYNTAX. This separation requires that EDIT, which assumes analysis always begins with a sentence prediction, receive information from SYNTAX specifying which predictions were actually in the pool at the start of analysis. Unfortunately, to provide for either of these requirements, period prediction or communication from SYNTAX to EDIT, necessitates extensive, complicated reprogramming.

A second possible procedure for analysis of incomplete titles is to modify the analyzer grammar so that relative to it, noun phrases and pre-

positional phrases become well-formed input. This is achieved by constructing two new sets of grammar subrules, one set to permit acceptance of noun phrases and one set to permit acceptance of prepositional phrases. For example, one new subrule would be of the form:

Sentence,Preposition ⟶ Object+Period

Here the argument pair is 'Sentence,Preposition', and the set of new predictions replacing 'sentence' in the prediction pool consists of an object prediction and a period prediction.

Unfortunately the addition of such rules brings with it one very undesirable side effect. When the analyzer is used to parse complete sentences, a single noun phrase or prepositional phrase is in fact ill-formed relative to usual rules of sentence construction. Experience with multiple-path predictive systems for Russian[4] and English has shown that rules of the type to be added for noun phrase and prepositional phrase analysis will, when the input does consist of complete sentences, tend to produce a number of parsings which are semantically or syntactically absurd. Thus although an assumption of well-formation is made for all input sentences, rules which would accept ill-formed input can have undesirable effects on the analysis of well-formed sentences. Since in the SMART analysis programs only the first analysis is used, the possibility of this analysis being an incorrect parsing should be minimized.

One solution is to make the analyzer selective as to the available grammar rules at the start on analysis. A first pass through the input would

be made with the usual grammar subrules active, a second pass with the noun phrase rules added, and a final third pass using the prepositional phrase rules. In effect the analyzer would have access to three separate grammars, each with its own criteria for well-formation. Successive grammars would be used only if previous ones were unsuccessful. A necessary assumption here is that the sole possible reason for failure of the first analysis pass is that the input is in fact an incomplete sentence, which in turn implies that the usual grammar is able to handle all complete sentences. Since the structural complexity of titles is limited, all titles which are also complete sentences should be analyzed successfully, and the assumption will be valid. Moreover, when the analyzer is used for textual analysis, where complete sentences are assumed, the probability of the second and third tries, if used, giving an analysis is negligible, because of the absence of a predicate prediction of any kind in the rules added for noun phrase and prepositional phrase analysis. Thus a single modified analyzer would be sufficient for both title and text analysis.

4. Analyzer Modifications

After considering the relative merits of the two methods discussed in Part 2, it was decided that the second, grammar modification, was the more feasible of the two, as well as being theoretically more satisfying. Implementation of the changes to the grammar began with the writing of the two new sets of grammar subrules; sample subrules from these sets are given in Fig. 1. Construction was straightforward, new rules being derived from a comparison of the noun or prepositional phrase structures already accepted

ARGUMENT PAIR	SR	AGREE TEST	NEW PREDS	MNEMONIC DESCRIPTIONS OF PREDICTIONS	STRUCT, SHIFT CD	ENGLISH EXAMPLES
Noun Phrase Subrules						
SE,NNN-9	SV	00001		SENTENCE	1S	COMPUTERS
			PD-	PERIOD	0 1.	.
SE,NNN-B	SV	00001		SENTENCE	1S	COMPUTERS
			AC-	ADJECTIVE CLAUSE	2 1S7S (1S7V) (1S7C)	THAT ARE HUMAN
			PD-	PERIOD	0 1.	.
SE,NNN-C	SV	00001		SENTENCE	1S	COMPUTERS
			XD-A	(A) AND (B)	1 1+	AND
			MC-A	NOUN SUBJECT	1 1S	AUTOMATION
			PD-	PERIOD	0 1.	.
SE,NOU-0	SV	00001		SENTENCE	1SA	STUDENT
			7C-A	SUBJECT MASTER	1 1S	ASSOCIATIONS
			PD-	PERIOD	0 1.	.
SE,NOU-1	SV	00001		SENTENCE	1SA	COMMUNICATION
			CN-D	COMMA	2 1S,	,
			A1-A	ATTRIBUTIVE ADJ	1 1SA (1S+) (1SA)	ELECTRONIC AND ASTRONAUTICAL
			4C-A	MODIFIED SUBJECT	1 1S	COMPANIES
			PD-	PERIOD	0 1.	.
Prepositional Phrase Subrules						
SE,PRE-6	PH	00010		SENTENCE	1PR	ON
			NQ-G	NOUN OBJECT	2 1PO	COMPUTERS
			ZC-E	(A,B,) AND (C) (DROP)	1 1+	AND
			DA-	ADVERB	1 1PR (1PO)	ON (THEIR) USES
			PD-	PERIOD	0 1.	.
SE,PRE-7	PH	00010		SENTENCE	1PR	ON
			GR-B	GERUND	2 1POG	COMPUTING
			ZC-E	(A,B,) AND (C) (DROP)	1 1+	AND
			DA-	ADVERB	1 1PR (1PO)	ON (ITS) USES
			PD-	PERIOD	0 1.	.

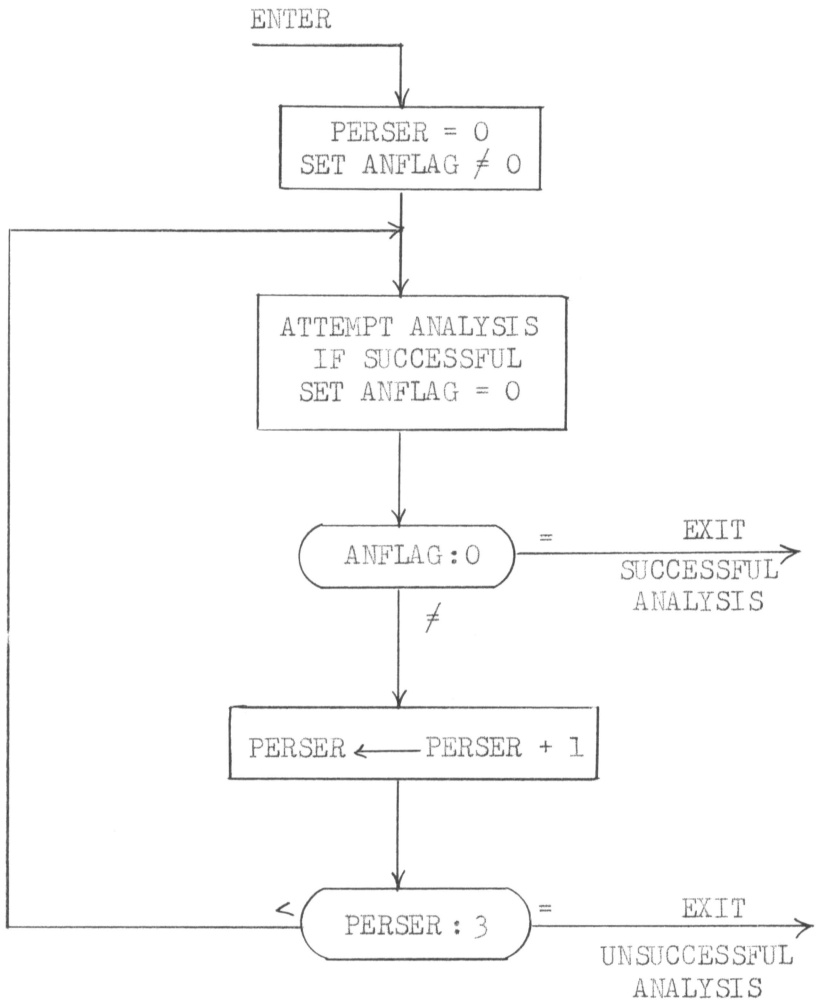
Sample Subrules for Incomplete Sentences

Figure 1

by the grammar with the possible noun and prepositional phrase structures occurring in titles. The titles examined were those from the SMART document collection, numbering 500 documents.

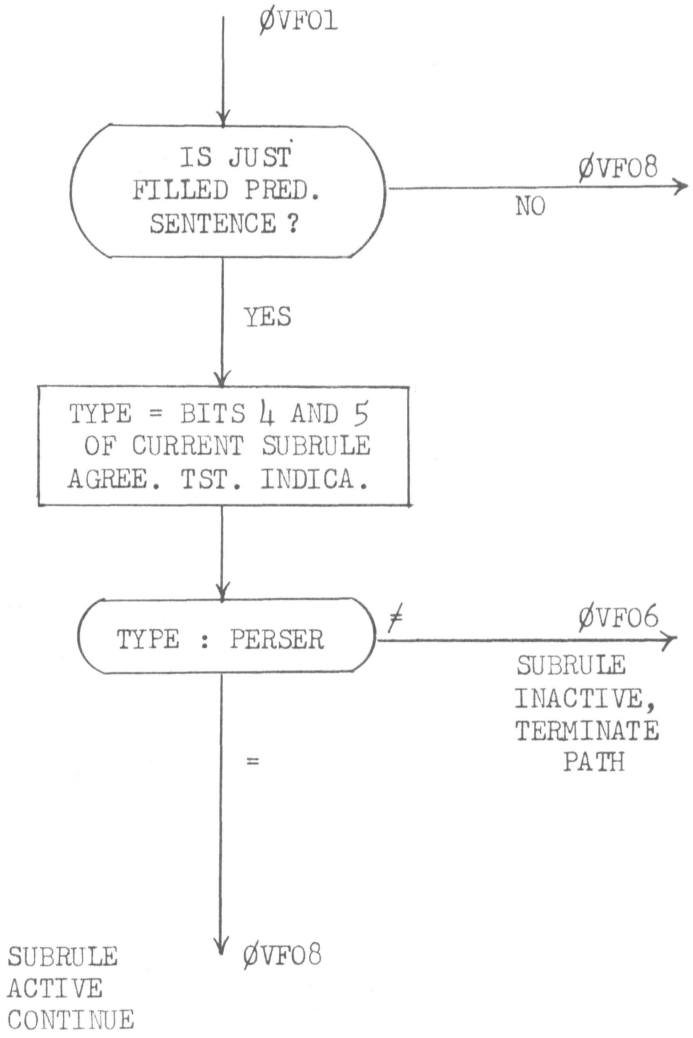
Since all the new grammar subrules would be indexed by an argument pair whose first member was 'sentence', the problem of activation of the subrule sets was reduced to consideration of only those subrules already in the grammar which were also indexed by a 'Sentence,xxxx' argument pair. Moreover, a savings in analysis time could be had if the additional rules were not simply added to the existing grammar for the second and third passes, but instead were substituted for the usual 'Sentence,xxxx' rules when the first analysis attempt failed. To this end a tagging scheme was devised which divides the 'Sentence,xxxx' subrules into three sets: complete sentence subrules, noun phrase subrules, and prepositional phrase subrules. For each pass through the input sentence only one set is permitted to be active. The result is equivalent to using three different grammars.

Tagging of the subrules was accomplished by means of the two rightmost bits of the five bit "Agreement Test" indicator which is part of every subrule in the grammar. For reasons which are not relevant here, these two bits are always zero in a 'Sentence,xxxx' subrule, and could therefore be unambiguously redefined for the new information. An added bonus is obtained from their use, for the analyzer does not make reference to them if the subrule under consideration is indexed by a sentence prediction; thus no special bypasses had to be programmed. The bit assignments are 00 for regular subrules, 01 for noun phrase subrules and 10 for prepositional



Over-all Flow of Modified SYNTAX

Figure 2



Determination in ØVFLW of Active Rules

Figure 3

phrase subrules. In reading this modified version of the SMART grammar, it is therefore necessary to interpret the agreement test bits differently for subrules 'Sentence,xxxx'.

Figure 2 is a flowchart of the over-all operation of the modified SYNTAX program of the analyzer. PERSER (permissible sentence rules) is a flag used to count passes and to determine which rules are active on any pass. This determination is carried out in the section of SYNTAX called OVFLØW (Fig. 3),† and occurs immediately after a prediction has been fulfilled. If this prediction is 'sentence', the grammar subrule indexed by the fulfilled prediction and homograph is obtained. Bits four and five of the agreement test indicator are algebraically compared with the current value of PERSER. Equality denotes that the subrule is active and analysis continues; otherwise the path is terminated. If during any pass the end of the sentence is reached ANFLAG is set to zero. Before PERSER is incremented ANFLAG is checked for successful analysis, which terminates processing of this particular input sentence. Three passes are made, each set of 'Sentence,xxxx' rules being tried once.

---

† This diagram should be used with the complete analyzer flowcharts.[5]

THE FOLLOWING ANALYSES WERE MADE USING ENGLISH GRAMMAR AM221A PREDICTION POOL SIZE = 100 NESTER MAXIMUM = 8

ANALYSES OF SENTENCE NUMBER 000001

WORD	HOMOGRAPHS				
WHAT	RL3	RL4	IPO	IPN	IAD
ARE	BE1P	BE2P	BE3P		
COMPUTERS	NNNP	PPPP	NOUP		
=	QUE				

***** ANALYSIS NUMBER 1 OF SENTENCE NUMBER 000001

ENGLISH	SENTENCE STRUCTURE	SWC	SWC CODE	SYNTACTIC ROLE	RL NUM	PREDICTION POOL
WHAT	2C	IPN	INTERROG PRN NOM	COMPLEMENT OF PRED VERB	SEIPN1	SE
ARE	2V	BE2P	BE2-COPULA	PREDICATE VERB	EXBE20	QU 1ZAEZB
COMPUTERS	2S	NOUP	NOUN 1	SUBJECT OF PREDICATE VERB	1XNNNO	QU 1PA
=	2=	QUE	QUESTION MARK	END OF SENTENCE	QUQUEO	QU

POOL OVERFLOWS= 0 NUMBER TEST FAILURES= 0 SHAPER OVERFLOWS= 0 NESTER OVERFLOWS= 0 TIME= 0.0 MINUTES

Sample of Analyzer Output for "What are Computers?"

Figure 4

ANALYSES OF SENTENCE NUMBER 000002

WORD	HOMOGRAPHS
ON	PRE AV2
COMPUTERS	NNNP MMMP NOUP
	PRD

***** ANALYSIS NUMBER 1 OF SENTENCE NUMBER 000002

ENGLISH	SENTENCE STRUCTURE	SWC	SWC CODE	SYNTACTIC ROLE	RL NUM	PREDICTION POOL
ON	1PR	PRE	PREPOSITION	PREPOSITION	SEPRE6	SE
COMPUTERS	1PO	NOUP	NOUN 1	OBJECT OF PREPOSITION	NQNNNO	PD NQG
.	1.	PRD	PERIOD	END OF SENTENCE	PDPRDO	PD

POOL OVERFLOWS= 0 NUMBER TEST FAILURES= 0 SHAPER OVERFLOWS= 0 NESTER OVERFLOWS= 0 TIME= 0.0 MINUTES

Sample of Analyzer Output for "On Computers"

Figure 5

ANALYSES OF SENTENCE NUMBER 000003

WORD	HOMOGRAPHS
COMPUTERS	NNNP MMMP NOUP
.	PRD

***** ANALYSIS NUMBER 1 OF SENTENCE NUMBER 000003

ENGLISH	SENTENCE STRUCTURE	SWC	SWC CODE	SYNTACTIC ROLE	RL NUM	PREDICTION POOL
COMPUTERS	1S	NOUP	NOUN 1	SUBJECT OF PREDICATE VERB	SENNN9	SE
.	1.	PRD	PERIOD	END OF SENTENCE	PDPRD0	PD

POOL OVERFLOWS= 0 NUMBER TEST FAILURES= 0 SHAPER OVERFLOWS= 0 NESTER OVERFLOWS= 0 TIME= 0.0 MINUTES

Sample of Analyzer Output for "Computers"

Figure 6

REFERENCES

1. Information Storage and Retrieval, Report No. ISR-7 The Computation Laboratory of Harvard University (June 1964).

2. Mathematical Linguistics and Automatic Translation, Report No. NSF-8 The Computation Laboratory of Harvard University (January 1963).

3. Mathematical Linguistics and Automatic Translation, Report No. NSF-9 The Computation Laboratory of Harvard University (May 1963).

4. Plath, W. "Automatic Syntactic Analysis of Russian," Mathematical Linguistics and Automatic Translation, Report No. NSF-12 The Computation Laboratory of Harvard University (June 1963).

5. Hazel, I., Sherry, M., and Tukis, C. C. "The Revised Multiple-path Syntactic Analysis Programs for English," Mathematical Linguistics and Automatic Translation, Report No. NSF-13 The Computation Laboratory of Harvard University (March 1964), pp. II-23, II-39.