DOCUMENT RESUME

ED 336 955 FL 019 489

AUTHOR Liou, Hsien-Chin
TITLE Computer-Assisted Writing Revision: Development of a Grammar Checker.
PUB DATE 26 Mar 91
NOTE 23p.; Paper presented at the Annual Meeting of the Teachers of English to Speakers of Other Languages (25th, New York, NY, March 24-28, 1991).
PUB TYPE Reports - Descriptive (141) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; Computer Software; Dictionaries; *Discourse Analysis; Editing; English (Second Language); Error Analysis (Language); *Error Correction; *Grammar; Phrase Structure; Programing; *Revision (Written Composition)

ABSTRACT

A computerized grammar checker was developed to assist teachers of English as a Second Language in editing student compositions. The first stage of development consisted of an error analysis of 125 writing samples collected from students. The 1,659 errors found were classified into 14 main types and 93 subtypes. This analysis served as the basis for constructing a taxonomy of mistakes and ranking the categories according to frequency of occurrence and comprehensibility. The grammar checker was then designed with a small electronic dictionary containing 1,402 word stems and necessary features, and a suffix processor to accommodate morphosyntactic variants of each word stem. An augmented transition network parser equipped with phrase structure rules and error patterns was then constructed. In addition, a set of disambiguating rules for multiple word categories was designed to eliminate unlikely categories, increasing the parser's efficiency. The current implementation detects seven types of errors and provides corresponding feedback messages. Future research will focus on detecting more kinds of mistakes with greater precision and on providing appropriate editing strategies. (Author/MSE)

ED336955

Computer-Assisted Writing Revision: Development of a Grammar Checker[1]

Hsien-Chin Liou

Abstract: In order to leave more time for EFL teachers to work on higher-level re-writing tasks, we decided to develop a computer grammar checker. The first stage of development was devoted to error analysis of 125 writing samples collected from our students. We found 1659 errors and classified them into 14 main types and 93 subtypes. The analysis served as the basis for constructing a taxonomy of mistakes and ranking the categories according to frequency of occurrence and comprehensibility. To implement the grammar checker, we first built a small electronic dictionary with 1402 word stems and necessary features, and designed a suffix processor to accommodate morpho-syntactic variants of each word stem. We then constructed an ATN parser, equipped with phrase structure rules and error patterns. In addition, a set of disambiguating rules for multiple word categories was designed to eliminate unlikely categories and thus increase the parser's efficiency. The current implementation detects seven types of errors and provides corresponding feedback messages. Future research will be focused on detecting more types of mistakes with greater precision and on providing appropriate editing strategies.

Keywords: EFL, grammar checker, error analysis, error patterns, electronic dictionary, word features, suffix processor, phrase structure rules, parser, feedback.

I. Introduction

One of the reasons language teachers in Taiwan, R. O. C. find EFL (English-as-a-foreign-language) writing classes formidable is the seemingly endless task of correcting grammatical mistakes in student compositions. Our own experiences in this area led us to investigate computer-assisted language learning (CALL). If a computer program could help detect or even correct grammatical mistakes in students' papers, it would reduce the tiring part of revision process and leave more time for human teachers to work on higher-level re-writing tasks.

---

[1] The paper was presented at TESOL '91 (New York, March 26) and is a progress report of a research project sponsored by National Science Council (#NSC80-0301-H007-15) in Taiwan, Republic of China. I would like to acknowledge Dr. Von-Wun Soo's contribution of setting up the global design of on-line implementation for this research project.

We[2] began by testing the extent to which a commercial software package, *Grammatik IV* (Price, 1989), could help our EFL students[3]. We asked 28 college students to use *Grammatik IV* individually, observed how they responded to feedback messages the package generated, and requested them to fill out a questionnaire which elicited their affective reaction toward the process. Mistakes detected and marked by *Grammatik IV* were recorded on a hard copy of student essays. It was found that though most of the students were using CALL the first time, they did not find the experience discomfiting. Seventy percent found the process interesting, and the package easy to use as well as helpful for one or another aspect of the revision process. Nevertheless, comparison of the marked essays with the originals revealed that only fourteen percent (10 out of 70) of the mistakes *Grammatik IV* detected were substantive grammatical errors. The rest of them concerned mechanics and stylish suggestions, some of which have become not as rigid today (as also pointed out in Dobrin 1990). Worse, the package missed significant errors frequently made by students, and generated false positives and misleading messages such as those in brackets below:

(1) *Having listening _ the teachers' word, I was not surprised at the poor score I got as I didn't do the question with caution.* [Passive voice: 'was surprised' Consider revising using active]

(2) *There <u>were</u> great man in the world whom I respected forever.* [The context of 'whom' indicates you may need to use 'who']

(3) *These occupy successively lower vanges on the scale of computer translation ambition.* [Usually 'these' should be followed by a plural noun.]

The failure in *Grammatik IV* is due to either erroneous analysis of sentence structures or rigid conformity to rhetorical conventions. Furthermore, because the package is designed for native speakers of English (L1), some suggestions for writing styles or word usage are not useful for our students with limited English writing proficiency, some of whom still have great difficulties with basic English structures. *Grammatik IV* deficiencies, thus, led us to try to develop an automatic English grammar checker which could detect the kinds of major errors our students

[2] A research group, including a professor in foreign languages, a professor in computer science, two part-time graduate students, and a full-time research assistant.

[3] We also examined *Right Writer* (Rightsoft, 1988), but found it to be an inferior product for our purpose.

frequently make[4].

II. Error Analysis and Categorization

Error analysis is derived from a common belief held by applied linguists and computational linguists that (a) errors or interlanguage systems (a linguistic system emerges as a second/foreign language learner tries to acquire another language) are systematic (Corder, 1981); or (b) ill-formedness is rule-based in natural language understanding (Weischedel & Sondheimer, 1983). Two extensive studies (Chen, 1979; Chiang, 1981) on error analyses of composition written by EFL college learners with English majors in Taiwan have been conducted. Chen's analysis focused on syntactic errors, while Chiang's was much more extensive, including semantic and discoursal errors. While their goals were mainly for research and writing pedagogy, the current study aims to implement the results of error analysis on a computer program. In addition, the database we used, from students of various backgrounds, is much larger than theirs. The last difference is that the current project does not deal with misspellings that spelling checkers in commercial word processing packages have achieved to a very satisfying extent.

For this project, we collected over 1000 two-hundred-word compositions from students with mainly engineering backgrounds. For future testing of the grammar checker, we have typed 194 essays. In analyzing 125 of these, we found 1659 errors[5] which were classified into 14 major types (see Table 1).

---

[4] For a similar critique of *Grammatik IV*, see Brock 1990a and 1990b. Concurrent efforts such as Chen and Xu (1990) have been initiated, as complementary to the present research.

[5] We used a database package, dBASE III Plus to manage the error classification and the contexts, or sentences, where the error manifested.

Table 1

Major Types of Errors

I.	Verbs
II.	Nouns
III.	Adjectives
IV.	Adverbs
V.	Auxiliaries
VI.	Pronouns
VII.	Determiners
VIII.	Conjunctions
IX.	Prepositions
X.	Subject-verb/predicate concord
XI.	Lexicon
XII.	Form-Classes (part of speech)
XIII.	Sentence-level
XIV.	Mechanics

Each of the major types was then divided into several subtypes, a process which yielded 93 subtypes in total. For example, the subtypes under the major type verbs are listed in Table 2.

Table 2

Subtypes of Errors Under the Verb Category

V1 (be V; redundant be verb; double finite verbs)
People could contact with friends when they were lived away.

V2 (modal + past verb)
If you use it carefully, it could made many work for you.

V-sub (verb subcategorization errors)
They try their best to stop them happen_ again.

VT (wrong tense/aspect)
If the war happened, we can never live a good life.

VT-1 (verb tense disagreement between clauses)
If we were not interested in the basic research, then we will not go ahead any more.

VT-2 (tense disagreement in a compound)
...we must avoid hazardous by-product of science and utilized the good points of science.

VT-3 (tense disagreement at discourse level)
On holidays, I often went out of Taipei. I usually ride my motorcycle enjoying the speed of wind.

VT-4 (contracted form fails to show plurai form)
It's rainy last weekend.

VF (wrong verb forms -- parsive/progressive forms)
The classmates and the teacher are all keep in my mind.

To measure the gravity of the error types, we adopted two criteria: frequency of occurrence and level of comprehensibility. Frequency of occurrence was measured by dividing the number of occurrences of an error type by the total number of errors, 1695 (see Appendix A). To obtain a measure for the second criterion, level of comprehensibility, we asked two native English speakers (associate professors in linguistics) to grade examples taken from each subtype on a four-point scale. We then selected those categories which occurred more frequently, hindered comprehension more significantly, and could be processed by a grammar checker with relative ease and formulated them into error patterns for computational processing.

For patten matching, a reported project at the University of Pittsburgh (Hull, G., Ball, C., Fox, J. L., Levin, L., & McCutchen, D. 1987) has set an example for us. Though they targeted a similar enterprise, the project was designed for basic writers (again L1 users). While their use of pattern matching techniques helped shaping our project, the error types they focused were very different from ours in nature. For example, infinitive errors (to + past tense verb), homophone confusions (their and there), comma-be errors were seldom found in our students' papers. Reports on commercial packages are rare. Therefore, we have to formulate the error patterns unique in our students' papers.

For computer programs to recognize/detect errors, we either formulated errors into patterns or represented the errors as explicitly as possible. Here, three subtypes under the major type Verbs are taken as examples to illustrate how error patterns were formulated. They were coded as V1, V2, and V-sub respectively. To formulate the pattern for each error subtype, we pulled out all context fields of each error type from our database and examined how the errors were manifested. For instance, all the contexts of V1 errors are listed in Table 3.

Table 3

All Contexts of V1 Errors

Record#	context
29	... people could contract with their friends and daily when they were lived away.
68	... then many dangerous thing will be happened.
506	Scientists have done a lot of works which made our living pattern is different from those days.
639	... the earth would be die at last.
692	... although they are not necessary improve our material life directly.
782	Because the scient is progress too fast.
817	It is seem great for the results coming out from science.
833	Although science makes our lives more comfortable, is it all do good to us?
885	Science has occupied a part of our life, and we are enjoy the development and achievement that science bring to us.
911	... I was fortunately passed the entrance examination ....
964	All of them made the earth never be suitable to be lived.
1040	All my life was began to be contained in the textbooks....

The **V1** error patterns can be described as a be verb (optionally plus one or more words,) plus another non-be verb, which has a feature of [intransitive], or [transitive] followed by a noun phrase at the verb phrase level. To accommodate the exception as in record number 506 in Table 3 requires another pattern to describe: causative verb, make (optionally plus one or more words,) plus finite verb be. The pattern can be written formally in the following:

a' V[b] X - V[vi]

|_ V[vt] NP

b' V[c] X V[b]

(V[b]: be verbs; X: wildcard symbol; V[vi]: intransitive verbs; V[vt]: transitive verbs; NP: noun phrase; V[c]: causative verbs)

(Note: Tentatively X is defined as an arbitrary number of words.)

Second, the subtype of V2 error is a modal followed by an erroneous form of verb as in sentences (4) and (5).

(4) *If you use it carefully, it could made many work for you.*

(5) *We may divided our discussion into the following points.*

The error pattern for V2 can be described as:

modal V-ed/V-en

(read as a modal such as should, could followed by the past tense or past participial form of a verb).

Third, the subtype of V-sub concerns problems with verb subcategorization as you can see in sentences (6) and (7).

(6) *They try their best to stop them happen_ again.*

(7) *We can use convenient electrical equipments to help us doing many works.*

Referring to categorization in the framework of generalized phrase structure grammar (Gazdar, Klein, Pullum, & Sag, 1985), we classified English verbs into 33 categories according to their correct usage. Since we found that it is impossible to formulate error patterns for the V-sub type, we attempted to represent the correct patterns instead. The correct representation facilitates mapping of verb patterns of the erroneous input onto the correct representation.

Patterns such as these provided the basis for the error identification component described in section IV.

III. The Electronic Dictionary

For computers to structurally analyze words in natural English texts, we need an electronic dictionary. A survey of literature indicates that there are several comprehensive machine readable dictionaries available such as Longman Dictionary of Contemporary English, Webster's Seventh Collegiate Dictionary, Collins Bilingual Dictionary, and Collins Thesaurus (see Boguraev & Briscoe, 1987; Boguraev & Briscoe, 1989; Byrd, Calzolari, Chodorow, Klavans, Neff & Rizk, 1987 for examples). However, because our students have limited English vocabulary and the project is exploratory in nature, we decided to make a small dictionary on our own to meet the immediate needs. Our experiences with this small dictionary will help us to select crucial information and to determine efficient access methods when we adopt an electronic comprehensive dictionary in the future.

For our own dictionary, a program was written to extract word types from a sample of the analyzed student compositions and formed the core of our dictionary entries. There are currently 1402 entries in our dictionary, including proper nouns. Each word is attached with part-of-speech (or word category) information and necessary features. Note that we have selected only the more likely part-of-speech information which our learners use in their English writing; we have not encoded rare usage in our dictionary. The selection and ordering of word categories are intended to reflect frequency of occurrence for the usage of each word, yet this requires further lexicographic research. This selective approach means that more unknown words could be encountered in higher quality essays. However, the simplification strategy saves the memory space and increases the parser's efficiency. A sample of word categories and their affiliated features in the electronic dictionary is shown in Table 4.

Table 4

A Sample of Word Entries and Their Selected Features in the Dictionary

Noun: count/noncount; vowel/consonant in the initial phoneme (V/C) **Adjective**: single/multiple syllable (S/M); V/C **Adverb**: subcategories (8 classes); S/M; V/C **Verb**: subcategories (33 classes) **Pronoun**: singular/plural/both (S/P/B); person (1st, 2nd, 3rd); case (subject/object/possessive) **Determiner**: S/P/B

The entries in our dictionary are mainly stems of words, or headwords. To accommodate suffix changes of word stems, we designed a suffix processor as suggested in the EPISTLE text critiquing system (Heidorn, Jensen, Miller, Byrd & Chodorow, 1982) by adopting the concept called a distributional lexicon (Beale, 1987). The processor is equipped with information about (a) rules of changes concerning word categories (e.g. from verb to noun) or the inflectional features (e.g. from plural noun to singular noun), and (b) associated actions (e.g. omitting -s in a plural noun can reform a noun stem). By means of a search procedure to correlate rules and suffix changes between the variants and headwords, the suffix processor ensures that the dictionary can identify the following three types of morpho-syntactic variants of each corresponding headword built in the dictionary: (a) the inflectional suffixes such as -ing, -ed, -s (for both verbs and nouns), (b) the derivational suffixes such as -ly in happily (from happy), -ful in cheerful (from cheer), and (c) markers of comparative and superlative degrees, -er, -est (such as hotter, or fastest). In this way, our dictionary can cope with natural English texts without building all the derivations as respective entries in our dictionary. To increase the processing efficiency, we grouped the rules above so that when a word like getting is encountered, it is assigned to the -ing group. This can save the searching time among all the suffix rules. To cope with irregular forms of verbs, we have designed a table which lists the root form, and irregular changes of verbs. In this way, an irregular verb (for example, began) can be associated with its root (begin).

In addition, we plan to build up a phrase dictionary and a dictionary of common problematic words to cope with errors in, for instance, sentences (8) and (9).

(8) *The misuse of the science results to the terrible thing of the rest part of the earth.* (should be results in)

(9) *We know that science is effected to human life seriously.* (should be science affects human life seriously)

IV. Parsing and Error Detection

The error patterns obtained from the analysis in section II were classified into eight levels of processing, based on ease of manipulation by the computer or linguistic analysis, if applicable. The classification will be revised as we analyze more student essays, generalize more and finer error patterns, and encounter bottlenecks after implementing the error patterns on line.

(I) matching strings: For instance, the mistake in (10) can be easily detected when we simply search for the words 'Although/Though' and 'but'.

(10) Although my high school years were full of pressure, but I still found my ways to relax myself.

(II) matching strings and sets: For instance, the mistake in (11) can be detected when we search for the words No matter and a set of question words such as when, where, who.

(11) No matter eating, clothing, living, and walking, we rely on science.

(III) using the suffix processor to cope with errors related to a certain category of words: The technique can, for example, handle the problem of pluralizing uncountable nouns. After failing to match the word informations as in (12) in our dictionary, the suffix processor can be used to reform the stem information. Since the countability feature for information indicates that it is uncountable, we can detect the nature of its error: an uncountable noun should not have a plural form.

(12) We must depend on some instruments like radio, computer to receive informations.

(IV) incorporating information in the dictionary into string matching: For instance, the mistake in (13) can be detected by matching the word more and searching for part-of-speech information of the following word in the dictionary. During the latter process, the suffix processor is activated to attach the feature [simple] or [comparative] degree to the word. This corresponds to the error pattern, 'more' + comparative degree of adjective/adverb, and the debugger can flag this mistake.

(13) The weather becomes more hotter than before.

(V) looking the problem up in a dictionary for common problematic words or phrases: As mentioned before, some of the students' mistakes are related to a specific word or phrase. This phenomenon will lead to construction of a specific dictionary with the hope of detecting such types of errors more effectively. In addition to problematic words, resolution techniques for detection will be built in the dictionary. This approach may help solve some of semantic problems which are not very meaning-dependent such as (14). With the help of parsing (to be described shortly), the program can detect the mistake: misuse of an adjective for an adverb. With the special dictionary, the program enables specific diagnosis of a common error type, confusion between everyday and every day (because of very similar forms).

(14) A lot of people feel nervous everyday.

(VI) using syntactic parsing and pattern matching: This level will be explained in more detail shortly as it is the main mechanism by which the most of the implementation work has been accomplished.

(VII) using semantic processing: Most of the diction problems fall into this category. This will be a very challenging problem as the information conveyed in the essays of our corpus is not within a limited domain. We have not yet had a clear idea of how to cope with such problems.

(VIII) using discourse strategies: Some of the errors concerning the scope of discourse such as anaphora may be too complex to be resolved in this project; however, we will explore the possible directions for future study.

To structurally analyze the input text, a top-down parser was constructed. It was formulated in the augmented transition network (ATN) grammar (Woods, 1970). To increase its precision of analysis, a set of word category disambiguation (WCD) rules has been devised to pre-process multiple word categories of some input words. For example, if a word has two categories, verb and adjective, and it is preceded by a determiner and followed by a noun, then the category, adjective is chosen (such as <u>falling</u> in <u>the falling rock</u>). The rules cut down the possibility of multiple word categories, and reduce the number of ambiguous sentence structures as well as processing time.

For the parser to be able to debug grammatical errors (besides judging whether the sentence is grammatical or not), two types of information were included in the program: an expert model and a bug model. The expert model represents all the structural possibilities of correct sentences, whereas the bug model represents the error patterns we have formulated. For the expert model, a small segment of phrase structure rules by which we need to generate the structure of a correct sentence looks like the following.

S -> NP VP
NP -> (Det) (AP) N ({PP, S'})
AP -> (Det) ("more") A {PP, S'}
VP -> V (NP) ({NP, PP})
PP -> P NP
S' -> Comp S
(S: sentence; NP: noun phrase; VP: verb phrase; Det: determiner; AP: adjective phrase; N: noun; S': embedded sentence; A: adjective; V: verb; PP: prepositional phrase; P: preposition; Comp: complementizer; ( ): optional

symbol; {}: selectional symbol)

The bug model currently has three groups of error patterns: those manifested at noun phrase, verb phrase, and clause levels. Each of the groups is activated while the parser is analyzing/reconstructing its corresponding constituent. In addition, there are errors which occur infrequently and/or are idiosyncratic. For these cases, we plan to map the expert model onto the input sentence and to diagnose the problem by some devised heuristic. The dual-model mechanism we applied is similar to that described in Weischedel and Sondheimer (1983).

How does our grammar checker operate to detect a mistake? The following is a flow chart (see Figure 1) which demonstrates how the grammar checker processes each sentence and detects errors. First, our program allows regular English texts as its input and processes sentence by sentence. For each sentence, the program uses the binary search algorithm to locate each word in the dictionary. If the program finds the word, it then records all associated features of this word. If the program fails, it proceeds to search for the word in the irregular verb table. If it finds the irregular verb form and thus the root form, then it returns to the dictionary and obtain features of the root form as well. If the program still can not find the word at this stage, it activates the suffix processor to do morphological processing. Notice that the category of a word before morphological processing is unknown and the word does not exist in the dictionary. After the word is processed by the suffix processor, it may be reformed and obtain its category information from this process. If the program still fails at this stage, the word is recognized as an unknown one for our current system. Up to this stage, the word category/categories information and associated features of each word, except unknown ones, have been assigned. At the error detection level, i. e. after each word has been associated with category information, the program activates word category disambiguation (WCD) rules to cut down unlikely categories if a word has more than one category. After WCD processing, each sentence obtains a hypothetically correct combination of word categories to be processed by the parser. If the parser determines the sentence as grammatical, the program proceeds to the next sentence. If the sentence is determined as ungrammatical and detected by any of the error patterns, the program reports the error/feedback message and continues for the next sentence. If neither the parser nor pattern matching can determine the status of the input sentence, another combination, if any, of word categories is assigned to the sentence, and the program repeats the parsing/pattern-matching processing. After the program exhausts all the possible combinations of word categories but still can not determine the status of the sentence (grammatical or ungrammatical), then the sentence is determined unable to be understood by the checker/the current system. The operation of the grammar checker is basically an interaction between the parsing and the error pattern matching processes.

The trace in Table 5 illustrates how sentence (15) is diagnosed.

(15) No matter _ he say_, he like_ these job_.

Table 5

An Output Trace

==

Parse sentence : **No matter he say, he like these job.**

Searching in the dictionary ....

Using WCD-rules ....

Assigning category ....
no <av> matter <n> he <ppn> say <v> he <ppn> like <v> these <d> job <n>

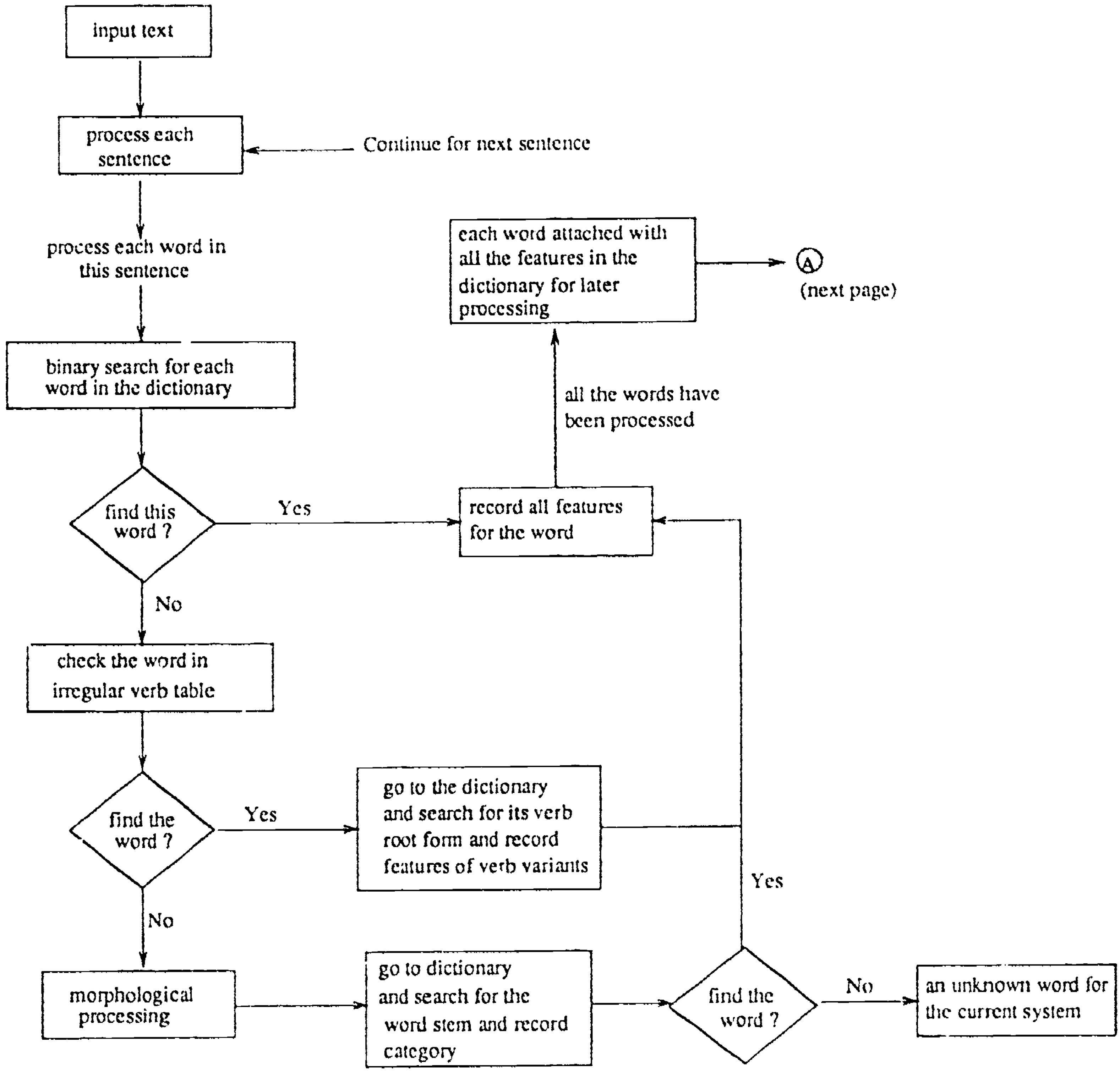
Syntax Error !! ---> No matter ......
no matter (?) he say, he like these job.

Syntax Error !! ---> Number disagreement: determiner -- noun
no matter he say, he like (these) (job).

Syntax Error !! ---> Subject-verb disagreement
no matter (he) (say), he like these job.
no matter he say, (he) (like) these job.

This is not a correct sentence. There are four errors.

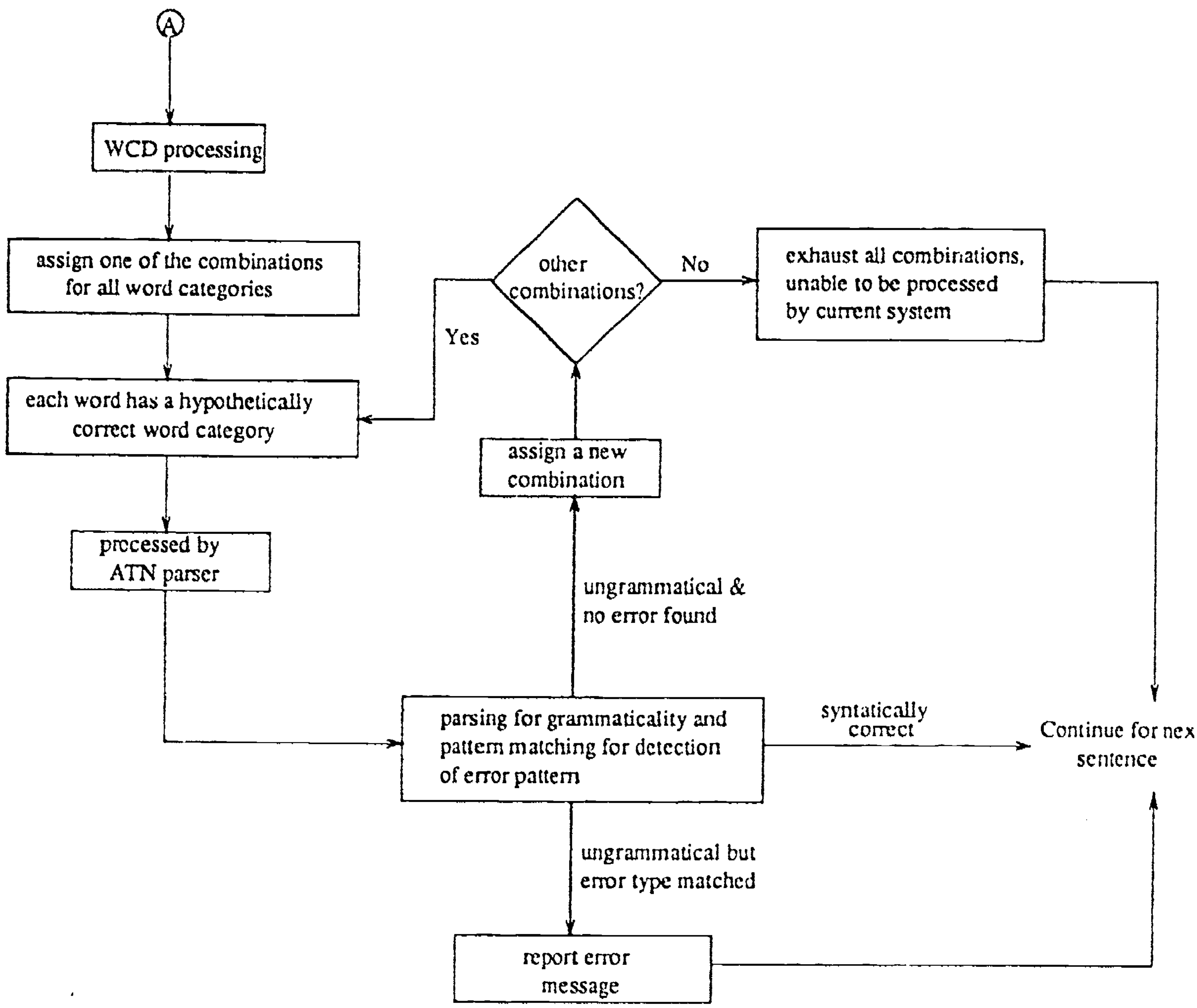
==



(to be continued)

Figure 1. The flowchart of processing a sentence in the program.

(to continue)


A
WCD processing
assign one of the combinations for all word categories
each word has a hypothetically correct word category
processed by ATN parser
other combinations?
Yes
No
exhaust all combinations, unable to be processed by current system
assign a new combination
ungrammatical & no error found
parsing for grammaticality and pattern matching for detection of error pattern
syntatically correct
Continue for next sentence
ungrammatical but error type matched
report error message

Because each sentence in the input is presumed to be ungrammatical, or erroneous, the program activates the error pattern matching process first. First, pattern matching of clause level errors is activated. Error types such as although ... but or no matter are classified as the clause level errors. This sentence matches the error pattern of no matter, a fact which the program notes. Second, since there is a noun phrase (NP), these job, error types at the noun phrase level are also examined; the phrase in question is found to match the type determiner-noun disagreement. Lastly, subject-verb (S-V) agreement is checked for each NP and VP (verb phrase) in each clause. The program first locates the head of each NP and VP and returns the number values (singular or plural) of both. Then, a comparison is made to see whether they agree. In sentence (15) two incidents of S-V disagreement are found. If none of the error patterns are matched in any of the constituents, the parser resumes its analyzing process to determine whether the sentence is grammatical under the current phrase structure representation in the program.

Currently, our checker can locate the following seven types of errors:

(I) although ... but combination

(16) Although he is poor, but he is happy.

(II) erroneous usage of no matter

(17) People can produce many things, no matter bad or good.

(III) determiner-noun disagreement

(18) We can know many informations.

(19) This is a books.

(20) I like an book[6]*.*

(IV) unbalanced coordinated phrases

(21) He likes a dog but hate_ a cat.

(V) capitalization misuse

(22) There are not the exist of Television, computer, airplane, and so on.

[6] The initial phoneme of book is encoded in the dictionary.

(VI) erroneous morphological changes in verb phrase, and

(23) I should went with you.

(VII) subject-verb disagreement

(24) Human create_ the science.

(25) Human already have the ability to research the phenomena of space.

(26) But the development in science have bring great change.

(27) A man who like_ art like_ books.

V. Feedback

When the program detects a grammatical error, appropriate feedback messages are essential for the grammar checker to achieve its educational goal. For this, we designed a message generating routine which basically matches a flag that is attached to each processing rule with a message file, and outputs the message to the users, possibly with some examples. We used a template to output a complete feedback message; namely, the message consists of some variables (as those underlined in (28)) and literal texts (those in plain texts in (20)). For example, a feedback message for sentence (28) is illustrated in the square brackets.

(28) The development in scientific technologies have bring great change.

[development is the subject of the verb have. The subject is in 3rd person singular form. The following are 2 correct examples:

The clerk beside the book shelves is watching television.

The lady who the workers love teaches English.]

For technical terms, we consider using Chinese. In addition, the correction and feedback given should be set up with a user-friendly interface environment so that language teachers and learners will not encounter confusion -- which may seem reasonable or common to computer-literate people, though.

Future Research and Implications

As an exploratory but ambitious research study, the current project has its drawbacks to be improved. Since we are aiming to treat the errors manifested in natural English texts, the coverage of English grammar, of both correct and incorrect ones, is much wider than much of the previous research work. Thus, the error detection tasks have been accomplished in an dissatisfying piecemeal manner. In the future, we plan to formulate the global mechanism of the grammar checker in a more generalized, from the linguistic perspective, framework. In addition, while the current project focuses on grammatical errors, we will take precaution to avoid posing prescriptive standard. One of the worst points in commercial grammar/style checkers is the prescriptive standard they try to reinforce. Commenting on the standards in English stylish usage, Dobrin (1990) concludes: "CorrecText [one of the best commercial packages Dobrin believes] ... may be inundating the user with false positives not merely because its syntactic analyses are necessarily limited, but also because the standards it purveys simply don't apply much of the time" (p. 77).

In the short term, we will complete the analysis of the remaining compositions and continue to develop the program to include more error patterns. In addition, the grammar checker's performance must be tested with corpus. Last, we will consider at which appropriate point in the program to give feedback messages and student editing strategies to improve writing revision. Clearly, there is still a long way to go. Nevertheless, despite the problems and difficulties, we believe we have made some important first steps.

In the present research project, studies of error analysis can provide many pedagogical implications as previous research suggests. The current project which uses a much larger corpus from learners with quite different backgrounds can provide significant insights for English teaching in Taiwan, especially for engineering majors. Establishment of the corpus in the computer can help future research in many perceivable aspects. The effort on formulating error types -- from pedagogical or linguistic perspectives -- into computer processable rule patterns will shed some light on cognitive science. The exploration of parsing strategies in this project will further research in the field of natural language processing and suggest possible resolutions related to semantic or pragmatic processing. Lastly, computer processing of erroneous natural texts made by foreign learners will pioneer the research in the fields of expert system and intelligent computer-assisted language instruction.

References

Beale, A. (1987). Towards a distributional lexicon. In R. Garside, G. Leech, & G. Sampson (Eds.), *The computational analysis of English* (pp. 149-162). London: Longman.

Boguraev, B., & Briscoe, T. (1987). Large lexicons for natural language processing: Utilising the grammar coding system of LDOCE. *Computational Linguistics, 13*(3-4), 203-218.

Boguraev, B., Briscoe, T. (Eds.). (1989). *Computational lexicography for natural language processing*. London: Longman.

Brock, M. N. (1990a). Customizing a computerized text analyzer for ESL writers: Cost versus gain. *CALICO Journal, 8*(2), 51-60.

Brock, M. N. (1990b). Can the computer tutor? An analysis of a disk-based text analyzer. *System, 18*(3), 351-359.

Byrd, R. J., Calzolari, N., Chodorow, M. S., Klavans, J. L., Neff, M. S., & Rizk, O. A. (1987). Tools and methods for computational lexicography. *Computational Linguistics, 13*(3-4), 219-240.

Chen, C. C. (1979). An error analysis of English compositions written by Chinese college students in Taiwan. Unpublished doctoral dissertation, University of Texas at Austin.

Chen, S., & Xu, L. (1990). Grammar-debugger: A parser for Chinese EFL learners. *CALICO Journal, 8*(2), 63-75.

Chiang, T. H. (1981). Error analysis: A study of errors made in written English by Chinese learners. Unpublished master's thesis, National Taiwan Normal University, Taipei.

Cook, V. (1988). Designing a BASIC parser for CALL. *CALICO Journal, 6*(1), 50-67.

Corder, S. P. (1981). *Error analysis and interlanguage*. Oxford University Press.

Dobrin, D. N. (1990) A new grammar checker. *Computers and the Humanities, 24*(1-2), 67-80.

Heidorn, G. E., Jensen, K., Miller, L. A., Byrd, R. J., & Chodorow, M. S. (1982). The EPISTLE text-critiquing system. *IBM Systems Journal, 21*(3), 305-326.

Hull, G., Ball, C., Fox, J. L., Levin, L., & McCutchen, D. (1987). Computer detection of errors in natural language texts: Some research on pattern matching. *Computers and the Humanities, 21*(2), 103-118.

Price, G. D. (1989). *Grammatik IV: User's guide*. San Francisco, CA: Reference Software International.

Rightsoft, Inc. (1988). *Right Writer: User's guide*. Sarasota, FL: Right Soft Incorporated.

Weischedel, R. M., & Sondheimer, N. K. (1983). Meta-rules as a basis for processing ill-formed input. *American Journal of Computational Linguistics, 9*(3-4), 161-177.

Author's Biodata

Hsien-Chin Liou took her Ph.D. at the University of Illinois. She is currently associate professor in Foreign Languages, National Tsing Hua University at Hsinchu, Taiwan, R. O. C. She teaches CALL, EFL Writing, and introductory Computational Linguistics. Her research interests include CALL for writing instruction, multimedia courseware development, and CALL research. She is a contributor of Computer Assisted Language Learning and Testing (ed. by Patricia Dunkel, 1991).

Author's Address

Hsien-Chin Liou
Foreign Languages, National Tsing Hua University
101, Sec. 2 Kuang-Fu Road
Hsinchu, Taiwan, R. O. C. 30043
E-mail: hcliu@fl.nthu.edu.tw
nthut048@twnmoe10.bitnet

APPENDIX A

Descending Distribution of Errors in Main Types and Subtypes

MAIN TYPE	N	PER CENT
Det	326	19.65 %
Verb	231	13.92 %
Noun	178	10.73 %
PS	174	10.49 %
Concord	168	10.13 %
Sent	158	9.52 %
Prep	123	7.41 %
Lex	115	6.93 %
Conj	67	4.04 %
Mech	54	3.25 %
Adv	27	1.63 %
Adj	23	1.39 %
Pron	9	0.54 %
Aux	6	0.36 %
Total	1659	100.00 %

TYPE	SUBTYPE	N	PER CENT
Det	A-3	154	9.28 %
Noun	CN	129	7.78 %
Det	A-1	105	6.33 %
Lex	Dict	94	5.67 %
Prep	Prep-1	81	4.88 %
Concord	3S-1	75	4.52 %
Sent	Run-on	65	3.92 %
PS	PS-nadj	65	3.92 %
Verb	V-sub	59	3.56 %
Sent	Frag	57	3.44 %
Verb	VT-1	55	3.32 %
Conj	Conj-1	55	3.32 %
Verb	VT-3	51	3.07 %
Mech	Cap	49	2.95 %
Det	Det-a	49	2.95 %
PS	PS-adjn	41	2.47 %
Verb	VF	39	2.35 %
Concord	SV	39	2.35 %
Noun	UN	27	1.63 %
Adj	Comp-1	23	1.39 %
Prep	Prep-2	23	1.39 %
Concord	3S-4	21	1.27 %
Noun	NN	20	1.21 %
Prep	Prep-3	19	1.15 %
Concord	3S-5	15	0.90 %
PS	PS-nv	14	0.84 %
PS	PS-adjadv	12	0.72 %
Verb	V1	12	0.72 %
PS	PS-advadj	12	0.72 %
Det	A-2	9	0.54 %
Sent	E	9	0.54 %
PS	PS-vn	8	0.48 %
Concord	3S/paral	8	0.48 %
Adv	ED	8	0.48 %
Sent	2S	8	0.48 %
Sent	Paral	8	0.48 %
Conj	NM	7	0.42 %
Verb	VT-2	7	0.42 %

(to be continued)

(to continue)

TYPE	SUBTYPE	N	PER CENT
Pron	Pron-1	7	0.42 %
Adv	Adv-2	6	0.36 %
Concord	SP	5	0.30 %
Conj	AB	5	0.30 %
PS	PS-vadj	5	0.30 %
Adv	OS	5	0.30 %
Sent	Rel-1	5	0.30 %
Verb	V2	4	0.24 %
Aux	Aux-to	4	0.24 %
PS	PS-prepv	4	0.24 %
Det	Det-O	4	0.24 %
Lex	Dict-v	4	0.24 %
Lex	2V-1	4	0.24 %
Det	A-4	3	0.18 %
Adv	ASP	3	0.18 %
Mech	Ap	3	0.18 %
Lex	Dict-p	2	0.12 %
Verb	VT-4	2	0.12 %
Lex	Red	2	0.12 %
Sent	WH	2	0.12 %
PS	PS-adjv	2	0.12 %
Concord	3S-2	2	0.12 %
PS	PS-advconj	2	0.12 %
Adv	very/much	2	0.12 %
Verb	VT	2	0.12 %
Sent	Rel-3	2	0.12 %
Noun	One-N	2	0.12 %
Pron	anaf	2	0.12 %
Lex	SM	2	0.12 %
Concord	3S-3	2	0.12 %
Mech	Punct	2	0.12 %
PS	PS-conjprep	2	0.12 %
PS	PS-nadv	2	0.12 %
Lex	Sem-1	1	0.06 %
PS	PS-prepconj	1	0.06 %
PS	PS-N.PP	1	0.06 %
Lex	to/too	1	0.06 %
Lex	Dict-Es	1	0.06 %
Lex	A/E	1	0.06 %
Det	some/any	1	0.06 %
Det	Num-a	1	0.06 %
Concord	WS	1	0.06 %
Sent	Rel-2	1	0.06 %
PS	PS-infprep	1	0.06 %
PS	Red-Comp	1	0.06 %
Lex	PH	1	0.06 %
Sent	WHi	1	0.06 %
PS	N-adj	1	0.06 %
Aux	Aux-2	1	0.06 %
Aux	Aux-1	1	0.06 %
Lex	Dict-mb	1	0.06 %
Lex	Dict-e	1	0.06 %
Adv	TA	1	0.06 %
Adv	SA	1	0.06 %
Adv	Adv-1	1	0.06 %
Total		1659	100.00 %