A DIAGNOSTIC APPROACH TO THE DETECTION OF SYNTACTIC ERRORS IN ENGLISH FOR NON-NATIVE SPEAKERS

Mari Ikeda Bowden and Richard K. Fox

Abstract

Typical grammar checking software use some form of natural language parsing technologies in order to determine if an ungrammatical error exists. Each time when ungrammaticality is detected, the grammar checker seeks for what error it is and provides a single explanation. It is very possible and likely, however, that non-native speakers of English make multiple errors in one sentence. Those multiple errors are reported one at a time. Furthermore, a typical grammar checker often fails to detect syntactic errors that are unique to and common among non-native English speakers. It even sometimes misdiagnoses errors. This report presents GRADES, a diagnostic program that detects and explains grammatical errors made by non-native English speakers. GRADES performs its diagnostic task through the application of classification and pattern matching rules instead of through parsing. It searches for the verb-related errors and the noun-related errors one by one in one sentence by narrowing down the classification of the errors. All the explanations of the detected errors are given at the end of the process. This makes the diagnostic process more efficient and helps provide better explanations of errors for non-native English speakers than other grammar checkers. GRADES is also a demonstration that grammar checking can be performed without solely relying on parsing technologies.

INTRODUCTION

In second language acquisition by adults (here, defined over 18 years of age), learners invariably make errors. Having knowledge of grammatical rules of the learned language does not guarantee the production of acceptable language. To non-native speakers, it is not always easy to manage grammatical rules and monitor their outcome. It would be ideal that errors they have made are detected and provided assistance for their correction. This, however, would not be likely to happen except in the instructional setting. Even in the classroom setting, it would be impossible for individual learners to receive assistance from the instructor with their errors particularly when the production takes place in a written form. One possible solution for providing tutoring assistance for individual learners may be the grammar checker. The learner can simultaneously engage language production and error correction. Wong (1996) says in her paper that it is generally believed that the learner acquires grammatical rules faster and more effectively if he/she focuses on and learns from his/her own grammatical errors. If so, as Wong points out, the grammar checker certainly is a great tool to help the learner become more aware of how he/she uses the language. The grammar checker would help the learner eventually monitor his/her own language production successfully.

Unfortunately, however, no grammar checker is built to detect non-native speakers' errors. Natural language processing used in the grammar checker is originally intended to understand not languages by non-native speakers but natural languages from native speakers. Grammar checkers fail to detect non-native speakers' errors that are often observed, such as "I no can swim" or "I know very well him". Furthermore, because non-native speakers may forget to apply grammatical rules, it is very likely that multiple grammatical rules are violated, resulting in multiple grammatical errors in one sentence. If these errors are not detected, then there is no explanation of the cause of the errors. What is worse is that there may be a potential misdiagnosis as the result of undetected errors. For example, the sentence "He not work today" sometimes is misdiagnosed as a Subject-Verb disagreement instead of the lack of the auxiliary verb "does" for negation.

This paper reports on GRADES (GRAmmar Diagnostic Expert System), a system to tackle part of this problem. GRADES is an expert system to identify and diagnose common syntactic errors among non-native speakers, particularly Japanese-speaking learners of English. It does structural analysis for error detection, but the analysis is done differently from the traditional method used in the grammar checking and natural language processing. When GRADES checks on each error type, it takes into considerations how non-native speakers of English use the language. Because GRADES goal is simply the diagnosis of non-native speakers' errors, it is intended to detect them more successfully and provide more precise explanations of the cause of the grammatical error.

SYNTACTIC PROCESSING AND ITS CHALLENGE

Grammar checkers check the grammatical structure of sentences based on morphological processing and syntactic processing. These two steps are part of natural language processing intending to understand natural languages. Morphological processing is the step where individual words are analyzed into their components and non-word tokens, such as punctuation, are separated from the words (Rich and Knight 1991). Syntactic processing is the analysis where linear sequences of words are transformed into structures that show grammatical relationships between the words in the sentence (Rich and Knight 1991). This section will briefly explain how syntactic processing is performed.

Any language has rules on how words must be put together as a meaningful unit. Inside a sentence, some words go with each other more cohesively than others. These units are called constituents. Applying correct syntactic rules or understanding constituents in a sentence is intuition shared by all native speakers of any language, who pertain this language processing often without being aware of it. This process of recognizing the sentence structure and building its representation is called parsing. When computers do this task, they do this in two very different ways: top-down parsing or bottom-up parsing (Smith 1991).

Top-down parsing (Recursive Transition Network)

This process begins with a rule for sentence, then proceeds through the hierarchy of constituents (phrases and lexical categories), generating more specific phrases until rules are applied that translate categories into words. For instance, from a start point, the subject has to be sought. Because the subject consists of the noun phrase (NP) — a unit consisting of a noun or a noun and its modifier(s), the NP is now sought. The NP phrase network starts looking for a determiner or adjective. If it finds a determiner, it searches adjective or noun next. Once NP is identified, then it begins looking for phrases for the verb phrase (VP) — predicate of the sentence stating about the subject. If all the phrase networks are translated into words when the end of the sentence is reached, the sentence is thought to be parsed correctly. Since the operations proceed based on the phrase structure rules, this is a rule-driven approach.

Bottom-up parsing (Unification)

This parsing begins with words and their lexical categories, and combines them into higher-level phrases until the unit finally becomes a sentence. One method of this type employs a shift-reduce parser, which pushes a new unidentified lexical item onto the stack and reduces some top most items into a higher-level phrase if there is any rule of the grammar that tells that the top most items can make a larger phrase. For example, when there is a determiner in the stack and system encounters a noun, the noun is pushed into the stack and then, the combination 'determiner + noun' on the top of the stack is replaced with NP using the rule. The same operation is repeated and the parse will succeed when the only item left on the stack is an S(entence). In this case, since the operations proceed based on a word the system encounters, this is a data-driven approach.

The biggest challenge of parsing lies in the fact that language is an example of one-to-multiple relationships in most cases. This can be observed at any level of the language structure. For instance, the word "hand" can be a noun or verb. In the case where "hand" is a noun, the word has multiple meanings. One syntactic phrase can have multiple structures. For example, in NP the adjective or the noun can follow the determiner. The determiner may not even be present if a plural noun is used and it does not refer to something specific. What this means is that, at some point in either the top-down or bottom-up approach, it is often the case that more than one rule is applicable. Yet, the system has to make its decision. If it finds out sometime later that the selection was wrong, it needs to backtrack to that decision point and make a different choice. If the second choice is wrong, the same process is repeated again. The process is repeated over and over until the correct choice is finally made. Therefore, any parsing system needs to employ some strategy to avoid or minimize this backtracking process. Some of the often seen approaches to this problem are to create well-formed phrase lists and use partial parsing (Smith 1991, Allen 1987).

Well-formed phrase lists

The well-formed phrase lists contain phrases that have completed a network. For instance, in the sentence "A small dog was playing in the park", when the word "dog" is pushed onto the stack, the word is recognized as a noun. Because the noun is the last lexical category expected in the NP network, the network exits at this point. This means that the phrase "a small dog" has finished being parsed and it is placed on the list. The phrases on this list do not need to be re-parsed again. Storing information on which phrase is already parsed saves backtracking for finding a correct attachment point later because the possible attachment points are the either front or end of the phrase.

Partial parsing

Partial parsing looks for certain fragments that can be reliably identified and produces partial parses. Reliable fragments include: (1) noun phrases from the determiner through premodifiers to the head, but not including postmodifiers such as prepositional phrases, (2) verb phrases consisting of auxiliary verb sequence, adverbials up to the main verb and (3) noun phrase consisting of a proper noun. The following diagram shows an example of full parse and partial parse for the sentence: *John had a party at a hotel* in a case where NP the task of partial parsing is to identify NP and VP

Full parse: (S (Pro-Subj John)(VP had (NP a party))(PP at (NP a hotel)).)
Partial parse: (Pro-Subj John)(VP had)(NP a party) at (NP a hotel).

In this case, partial parse only identifies noun phrases and the main verb of the sentence, leaving a preposition out of analysis, whereas full parse identifies the prepositional phrase along with the verb-object relationship, indicating the complete phrasal structure of the sentence.

The two approaches for parsing are top-down parsing (Recursive Net Work) and bottom-up parsing (Unification). Both approaches pose a problem with deteminacy because they need backtracking at some point of analysis. In the parser, well-formed phrase lists and partial parsing are used to improve the determinacy.

SYNTACTIC PROCESSING APPLICATIONS

One well-known application of syntactic parsing explained in the previous section is a system that performs error detection. The grammar checker is one example of such a system. It usually employs the parser that uses some bottom-up unification-based parser to identify grammatical errors in written input by the user. Such an error detection system is also used in computer assisted language learning (CALL) programs in the area of foreign language learning. Those programs aim to provide individual tutoring assistance for people who study a foreign language. They usually take written input from the language learner, find morphological or syntactic errors in the input and provide exercises to avoid the same type of errors in the future. Some of the recent CALL programs employ the latest natural language processing technologies which are frameworks similar to the technology used in the grammar checker. (Those programs are particularly called

intelligent CALL programs.) This section will take a brief look at what parsing approach is actually used in some intelligent CALL programs.

The diagram below shows a simplified functional diagram of a typical parsing component used in a language tutoring system (Sams 1995).

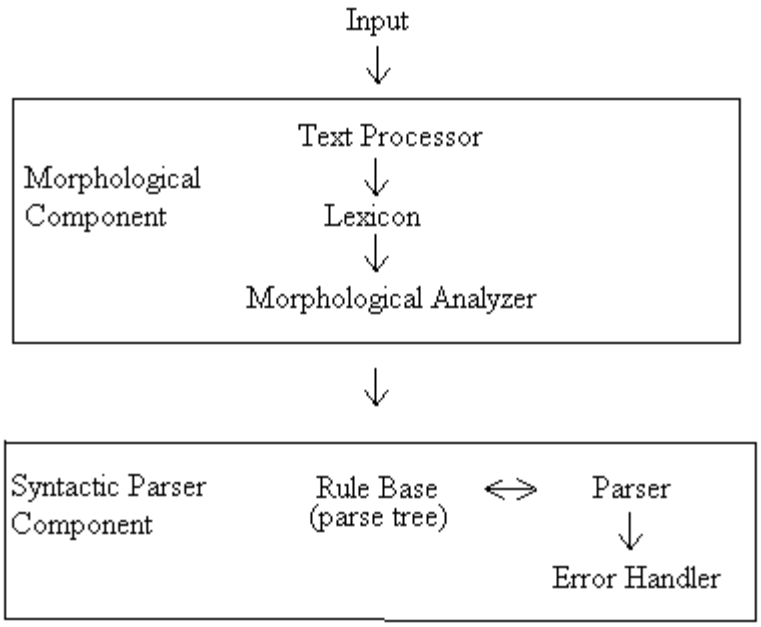


Figure 1: Mechanism of error detection used in CALL programs

The process starts with a sentence being sent to a text preprocessor and checked for spelling mistakes. It is then passed to the morphological analyzer, which decomposes the words into roots and affixes based on information from lexicon (e.g. part of speech, singular or plural). The roots and affixes are unified back into the original form and the sentence is sent to the syntactic parser. The parser tries to build its structure, called a parse tree, to locate syntactic errors. The common approach to parsing is the use of the bottom-up unification parser starting with the words and gradually building higher-level phrase structures until the S(entence) is formed. The parser applies the rules on sentence structure until it finds those that account for the sentence structure.

Although the unification method is popular, the unification parser alone may not be efficient. Therefore, some systems actually employ the top-down parsing strategies combined with bottom-up unification. For instance, GPARS (Lortiz 1995), which is a multilingual suite of computer-assisted language teaching programs, is built around Generalized Transition Network parsers. These parsers are syntax-centered, non-deterministic, right-branching, depth-first, top-down, and rule-driven, but at the same time, those strategies are intermixed with some strategies that increase determinism of the system. They include interleaved semantic processing and shift-reduce mechanism to reduce ambiguity as early as possible. Another strategy is the use of the ill-formed list along with the well-formed list. The Ill-formed phrase lists help the parser reject the phrases that have been already rejected during subsequent backtracking and reparsing.

It should be noted here that even though parsing is a syntactic operation, some semantic feature checking seems necessary even at the syntactic processing stage. A simple example of this occurs when selecting the kind of verbs in the progressive aspect in English (e.g. I *am cooking* now). Just because a word is a verb does not mean that it can be used with the progressive aspect. Action verbs like "run" or "laugh" can take the progressive form, but not stative verbs like "know" or psychological verbs like "love" and "hate" usually cannot. Semantic features of each verb definitely help the grammaticality judgement of the sentences containing the progressive aspect (e.g. "Chris is watching TV" vs. "*Chris is knowing Terry well". * signifies that the sentence is ungrammatical.)

It would be worth while mentioning that there is a view that using production rules alone may not be adequate for parsing ill-formed sentences. The traditional parsing method searches combinations of constituents that can be driven from a collection of production rules. Those rules are ones that native speakers use to produce their language. Natural language processing is originally intended to understand the language by revealing production rules that are embodied in sentences produced by native speakers. Input is, therefore, presumed to be correct. In contrast, in non-native speakers' production, those production rules are not always realized, which results in ill-formed sentences by those non-native speakers. DeSmelt (1995) doubts that applying the generative rules in analyzing presumably well-formed input to ill-formed input from the foreign language learners could do a sufficient job.

His system called Kerr Kommissar, which is a language learning system for intermediate German, uses predication-driven parsing instead of production-rules oriented parsing. The system first identifies the operating verb and then, if it is a transitive verb, it looks for a noun phrase that could function as its direct object with a certain syntactic features the verb mandates. If there is no noun that matches, the parser looks for the closest unassigned noun to meet the requirement.

Unification is a popular approach used in the parser, but top-down parsing combined with unification strategies is also used. The latter seems to be a more efficient approach. All of the parsers, no matter what parsing approach is used, use the production rules to parse the sequence of words in the sentence. As parsing proceeds, a parse tree is built to identify the violation of the production rules used at the time.

EXPERT SYSTEMS

This section will look at error detection from a different point of view. Error detection, particularly, in written input from non-native speakers can be viewed as diagnosis. Diagnostic systems are expert systems based essentially on search spaces of malfunctions or errors. This section will discuss how expert systems work to differentiate a diagnostic approach from the parsing-based approaches previously described.

Expert systems use a large amount of domain knowledge to solve a specific problem. The mechanism used by most expert systems is rule-based search. The search strategy used to conduct many diagnostic tasks is classification, where the characteristics of the super-class are shared by its sub-classes, and these child classes are exclusive to each other in terms of the presence or absence of a particular feature (Clancey 1985, Chandrasekaran 1988). In rule-based expert systems, a set of rules related to one category is used to narrow down the category in the form of classification. By applying relevant rules to a category, one of its sub-categories is selected to continue the further search.

In the case of error detection, there is a hierarchy of errors to be found. The input data is categorized in terms of the class of errors. For instance, in order to identify an error of the verb-related types, the verb phrase of the data is first identified. Then verb phrase is classified into a

more relevant class of the error by applying its rules, and other possible classes among the child nodes are ruled out. As long as the class of the error is relevant, this classification continues. If the error is found to be present or absence, classification stops that point. Figure 2 below illustrates the classification process explained above.

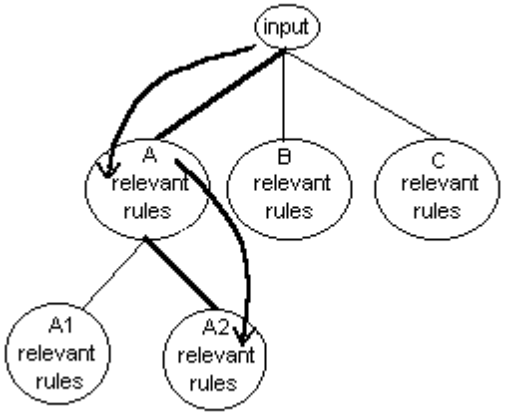


Figure 2: Hierarchical classification using rules

This way, the search for a solution using hierarchical classification (Chandrasekaran 1988) involves traversing the tree structure. In rule-based systems, the selection of the child nodes is determined by applying a group of rules that are related to the class of errors.

As is clear in the above explanation, the rule-based diagnostic approach works very differently from the parsing-based approach. The parsing-based approach starts with the input sentence, parsing it and gathering the information about the structure of the sentence. In contrast, the rule-based diagnostic approach starts with a list of errors. Related rules of the errors are grouped together and used to classify the data toward the identification of the errors. If an error class is found relevant, examination of all of the error's children classes take place in an attempt to refine the error into a more specific error. Diagnostic systems have been developed particularly in the medical/chemistry field or the business field. Many applications are found in the manufacturing/industrial setting. It seems that expert systems for grammar diagnosis have not yet been explored.

NON-NATIVE SPEAKERS' LANGUAGE

This section will turn the focus to non-native speakers' errors. Although non-native speakers of English have knowledge of grammatical rules, the knowledge does not necessarily get realized in their production, spoken or written. To native speakers of the language, many grammatical errors made by non-native speakers may seem to be merely the result of grammatical rules randomly dropped or incorrectly applied. Recent studies show that this is not entirely true. Many researchers have reported that some varieties of errors occur often among non-native speakers. This finding implies that non-native speakers' language is not consistent, and yet it still shares some characteristics and tendencies in terms of the types of errors. There are common varieties of errors

into which non-native speakers tend to fall. This section will briefly explain what those commonly observed varieties of errors are in order to show how differently non-native speakers use the language.

Based on other researchers' studies, Ellis (1986) writes that one type of the varieties is a linguistic situation where a certain grammatical violation occurs. He says that when a sentence consists of one subject and two (or more) verbs, even an advanced learner may omit 3rd person singular 's' with the 2nd verb and after. For example,

> He likes books and *buy* a lot of books each month.

Ellis also says that the learner may omit 3rd person singular 's' when two different syntactic structures fuse into one new sentence, such as:

> The man who *live* next door married yesterday.

This author has observed that 3rd person singular 's' tends to be dropped when an adverb or an inserted phrase is placed between the subject and the verb. For example,

> Mark sometimes *get* sick.
> My mother, as a professional writer, *travel* a lot.

Looking at the above examples, it seems to be the distance (that is, separation) between the subject and the verb that makes it harder to apply Subject-Verb agreement rule.

Another type of the variety of errors is of the incorrect syntactic structure. One collection of data presented here comes from chronological studies on second language development, also reported by Ellis (1986). According to him, there is conclusive evidence that there is some common pattern of error from basic to advanced level. The grammatical categories reported so far include negation and interrogation.

> **Negation**: Negative particle "no" may be used for "not". Negative particle "no/not" is located outside the verb phrase. If the auxiliary verb is present, it occurs before the auxiliary verb, and if no auxiliary verb is present, the form of "do" is not used (see also, Stauble 1984).
>
> > I *no can* swim.
> > I *not* work(ing) today.
>
> **Interrogation**: In the beginners' production, a wh-interrogative is simply added to a demonstrative sentence. In this stage, the form of "do" appears, but the form of "do" is often seen followed by the main verb like in the second example below. The main verb still carries the tense marker or 3rd person singular 's'. Embedded interrogative clauses have the structure as the independent interrogative sentence.
>
> > Where *you went*?
> > Who *did steal* his money?
> > Where *did* you *went* yesterday?
> > I don't know *what did she mean*.

The following are other types of errors that have been presented as often observed errors among non-native speakers in different linguistics or second language acquisition papers.

Articles: A definite or indefinite article is deleted when it is necessary (Robertson 2000; Towell 1994).

I have __ daughter.
Mike is in __ second grade.

Auxiliary verbs: The form of "be" is used for the form of *have* in the present perfect tense (Oshita 2000; Balcom 1997)

What *was(is)* happened?
He *was arrived* early.

In relation to the auxiliary verb for negation, it has been the author's observation that some beginners occasionally produce a sentence like " She is not have money" meaning "She does not have money". The speaker chooses an incorrect auxiliary verb for a negative sentence.

This section has intended to give the reader a picture of how non-native speakers of English use the language by presenting what errors are commonly found, and where and how they occur. Those data clearly show that a wide range of possible error types and syntactic structure has to be checked in order to detect non-speakers' grammatical errors.

GRADES (GRAmmar Diagnostic Expert System)

Types of errors sought

The systems that detect errors by using a parser do not begin to analyze sentences to detect particular errors. When input is parsed, any kind of syntactic features or rules that are recognized are checked. If the parser finds any violation of rules or discrepancy among the features, the error handler handles them for such violation by highlighting them, possibly supplying the violated rule or feature.

GRADES, on the other hand, is a goal-driven diagnostic system. The system starts with the goal of checking syntactic features for errors. It is not driven by either production rules or words used in input sentences but instead by a list of possible grammatical errors. The errors currently sought are the following seventeen types:

A. Verb-related errors (thirteen types)

- Subject-verb disagreement
- Subject-auxiliary verb disagreement
- Verbal elements out of order (auxiliary verbs and main verb)
- Incorrect verb form of the verbal elements (verb form requirement by an auxiliary verb immediately before)
- Incorrect verbal sub-structure of the verb (verb form when the verb is used right after the main verb, such as "avoid driving" vs. *"avoid to drive")
- Lack of the auxiliary verb "do" for the negation
- Incorrect choice of the auxiliary verb "be" with an intransitive verb
- Incorrect use of passive voice (the presence of verb "be" when not needed)

- Lack of the auxiliary verb "be" for the passive voice (the absence of verb "be" when needed)
- Lack of the main verb

"potential" errors
- Incorrect choice of the auxiliary verb "be" in the negation
- Lack of the auxiliary verb "be" for the progressive tense
- Lack of the auxiliary verb "have" for the present perfect

B. Noun-related errors (four types)
- Noun number disagreement in NP
- Lack of a determiner
- Extra determiner
- Modifiers out of order

In order to check subject-verb agreement, for instance, GRADES uses rules like the following.

If the subject is the 3rd person singular and the verb is the present tense,
then the verb has to take the 3rd person singular form by adding '(e)s' to its end.

If there is an auxiliary verb present,
then the main verb does not have to agree with the subject in number and person.

In order to check the verb form in the verbal sequence in the sentence, GRADES uses rules like the following:

The auxiliary verb or the verb after the modal auxiliaries has to take the original form.

The auxiliary verb or the verb after the auxiliary verb "have" has to take the past participle form.

Lexicon

GRADES has a lexicon of two hundred and nineteen (219) words. In GRADES, every inflected form of one lexical item is stored in the lexicon as separate entries. For instance, words "walk" and "walked" are treated as two separate lexical items even though they have the same root "walk". This simplifies implementation of the diagnostic rules.

Each part of speech forms a class containing lexical attributes. For example, the attributes of the auxiliary verb include:

(1) Category (modals, have, be or do)
(2) Verb form (original, present, past, past participle, present participle)
(3) Number requirement (used with a singular or plural subject)

GRADES uses the following ten parts of speech classes: adjective, auxiliary verb, determiner, noun, adverb, verb, personal pronoun, preposition, conjunction and punctuation. It is convenient to have adjectives and determiners to check the word order of noun modifiers. What discriminates determiners from adjectives is that determiners are a group of words that always have to come first among the modifiers preceding a modified noun (Frank 1972). Also, while multiple adjectives are

grammatical, multiple determiners except “all”, “both” and “half” are ungrammatical, for instance, “his the house” or “those our computers” (Frank 1972).

Adjective phrases of quantity, such as “a lot of” or “a few”, are not included in the lexicon.

Words are stored in the parts of speech lists corresponding to their part of speech. Some words are listed in multiple parts of speech lists.

Input

There is no restriction on the number of words used in input. Any number of words can be used in an input sentence as long as those words are in the lexicon. An input sentence is stored in the input sentence list. In this list, the lexicon of words contains each word’s speech roles. Some words may have multiple roles.

There is one requirement at the time of input. The user is required to specify the subject word by assigning the value 1.

Input restrictions

First, the sentences that are diagnosed are all simple sentences, which contain one subject and one predicate with a finite verb or finite verbs. For example,

My uncle loves books.
My uncle loves to write books.
My uncle loves books and buys many books every month.

Therefore, complex sentences and compound sentences, which contain more than two or more full predicates, cannot be diagnosed by GRADES. Compound sentences have two independent clauses combined with a co-ordinate conjunction (“and”, or, “but”), such as “ Mark goes to work, and his wife stays at home” (Frank 1972). Complex sentences have two or more clauses (Frank 1972). One of the clauses is an independent clause, and one or more of the clauses are dependent clauses. Some compound sentences a relative clause modifying a noun, such as “My uncle who lives in Texas loves books”. Others have two clauses joined with a sub-ordinate conjunction (e.g. “when”, “because”, “if”, or “although”), such as “When I looked up, I saw a spaceship in the sky”. At this point in its development, GRADES does not have rules to handle these more complicated structures although it is only a matter of adding more complex rules to handle such sentences.

Among simple sentences, the type of sentences that are diagnosed are statements and yes-no questions. For example,

I like oranges.
I did not sleep last night.
Did you call Mark?

Therefore, neither exclamation sentences (e.g. What a nice dress!) nor wh-questions (e.g. What time did you leave?) can be analyzed.

Second, the noun head of the NP has to only consist of one noun or pronoun. In other words, the words, such as “girls”, “boys” “he” or “I”, can be used individually as a subject, but if they become a noun or pronoun phrase combined by “and” or “or”, such as “boys and girls” or “he and I”, cannot be used in input. The gerund of a verb, such as “building” or “playing”, cannot be used, either.

Third, no inserted words or phrases, which are surrounded by commas, parentheses or dashes, can be used in input.

Lastly, no contracted forms must be used. Each word has to be spelt separately, such as "do not" instead of "don't" or "I have" instead of "I've".

ERROR DETECTION PROCESS IN GRADES

This section will first discuss the general structure of the error detection in GRADES in the first segment and how GRADES checks those seventeen types of errors previously introduced in the second segment.

1. General structure of error detection

GRADES detects two types of errors: verb-related errors and noun-related errors. Verb-related errors are sought first and next, noun-related errors.

Verb-related errors require the analysis of the verb phrase. Its first step is to identify the verb phrase of the input data. Easy as it is for the English speakers to visually recognize the VP, this identification process is not necessarily easy for computers because the VP is not always in a continuous sequence. There can be an adverb (phrase) or inserted phrases between auxiliary verbs. In questions, of course, the auxiliary verb is located before the subject. An obstructed sequence like this may affect the efficiency of the identification process. To make the search easier to handle, GRADES isolates auxiliary verbs and a verb that makes up the entire VP. It looks for auxiliary verbs or a verb in a sentence before and after the subject. It then takes out all the auxiliary verbs and the verb it could find, skipping other elements, and puts them in a separate list (called a verb element list) for the further operation.

Input sentence: He must have always thought about his work.

Verb element list: must have thought

During the process where GRADES looks for the entire the VP, it also checks three more pieces of information for later uses: if the negative particle "not" is present, if the auxiliary verb "do" is present before the subject (that is, the sentence is interrogation), and if the verb is followed by the NP. If the negative particle "not" is found, the Negation flag is set to 1. If the auxiliary verb "do" is found at the beginning of the sentence, the DoQuestion flag is set to 1. If the object noun is found, the VerbHasObject flag is set to 1.

Once the verb element list is set up, GRADES checks the types of verb elements in the list (auxiliary verb or verb) and their sequence pattern. Logical verb sequences are described later in the paper. During this process, a word order error among the verb elements is sought. If there is no word order error, GRADES identifies the sequence pattern of the verb elements in the list. Next, based on the sequence pattern, GRADES checks if each verb element takes the correct verb form in the particular sequence. During this process, verb form errors are detected.

Based on the category of the verb phrase pattern, other types of syntactic errors can now be checked. GRADES performs four types of syntactic testing at this stage:

1) whether the subject and verb/auxiliary verb agree in terms of person and number
2) whether the auxiliary verb "do" is present in a negative sentence
3) whether the passive voice structure " be + past participle" is syntactically a correct choice
4) whether the main verb is present when the form of "do" is used.

Figure 3 below illustrates the general structure of detecting verb-related errors by GRADES.

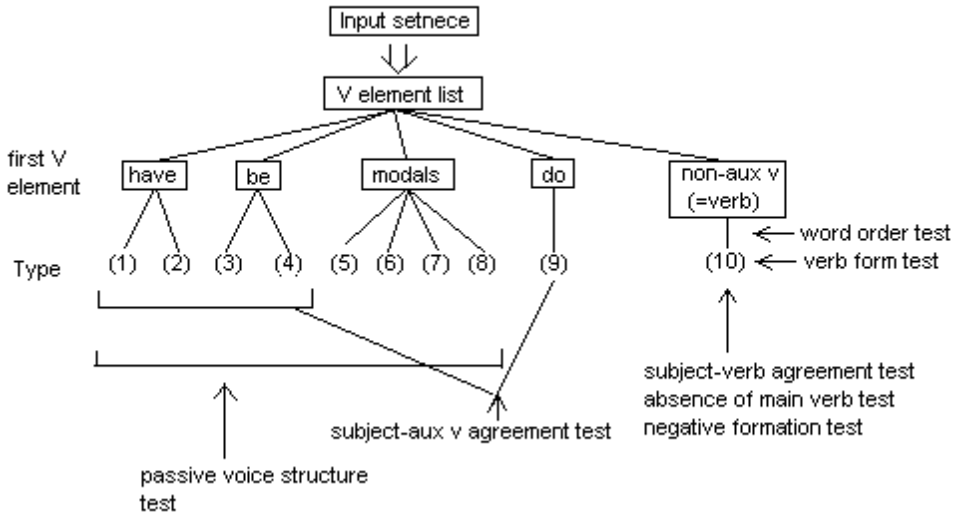


Figure 3: Classification of verb-related errors

The third layer immediately below the V element list indicates five classes into which the VP is categorized. The classification depends on the type of the 1st verb element in the VP. If there are auxiliary verbs, the 1st auxiliary verb can be "have", "be", modals, or "do". These types of auxiliary verbs will be described later. If there is no auxiliary verb used, the 1st verb element is a verb (that is, not an auxiliary verb). The next layer underneath shows how many possible sequence types there are among the auxiliary verbs in the case where the VP starts with the indicated verb element: "have", "be", modals, "do" or a verb. The AuxVP starting with "have" is categorized further into two types: (1) and (2), the AuxVP starting with "be" two types: (3) and (4), the AuxVP starting with "modals" four types: (5), (6), (7) and (8), the AuxVP starting with "do" one type: (9), and the VP only containing a verb has one type: (10). These types of the sequence of the auxiliary verbs will be explained in detail soon. The word order test, placed between the 3rd layer and the 4th layer, is performed during the process of analyzing the sequence of the verb elements. The verb form test underneath is applied to each category from (1) through (10) once it is identified. Then, subject-verb/subject-auxiliary verb agreement test and the four types of syntactic tests at the bottom of the tree are performed to the categories to which the arrow is pointing.

GRADES also uses rules for "potential" errors that provide another possible cause of the error. For instance, if the input is "The student would enjoyed the party", GRADES prints out an error message on the verb form "enjoyed" because the auxiliary verb "would" takes a verb in the original form. At the same time, it is pointed out that the lack of the auxiliary verb "have" may be another possible cause. Such rules are applied to certain categories identified as the result of verb elements sequence analysis.

GRADES now moves to detecting noun-related errors. With noun-related errors, GRADES first needs to recognize the NP. GRADES first looks for a noun in the sentence instead of

searching for a determiner or adjective, which is the case in parsing for the NP. As soon as it finds one, GRADES goes back in the sentence to see if the noun is preceded by a modifier (that is, determiner or adjective). It keeps checking until no modifiers are found. That means that (1) no more adjectives are found and no determiner is found, either or (2) no more adjectives are found but one determiner is found. During this process of identifying modifiers, a noun number disagreement in the NP is detected by comparing the number features between the noun and the modifier (if any). If a determiner is present, the DetPresent is set to 1. If at least one adjective is present the AdjPresent is set to 1 for later testing.

After the above process stops, GRADES searches one further word back from where it has stopped and checks the part of speech of this word to find if the determiner is correctly used. Detected errors include the lack of a determiner, extra determiner and ungrammatical determiner word order. Figure 4 below illustrates the general structure of detecting noun-related errors by GRADES.

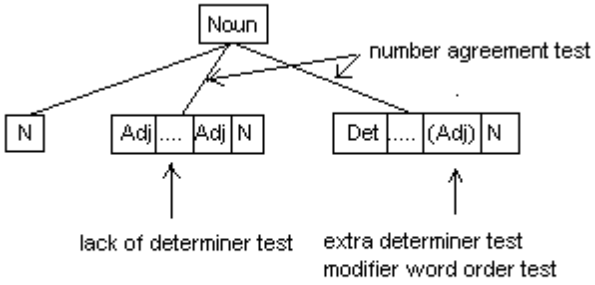


Figure 4: Classification of noun-related errors

The 2nd layer below the top "Noun" indicates the types of pre-modifiers to the noun: no modifiers, only adjective(s) and a determiner and/or adjective(s). The number agreement test, placed between the top layer and the 2nd layer, means that this rule is performed during the process of analyzing the modifiers. The three tests at the bottom of the tree are performed to the category to which the arrow is pointing.

2. Rules used for error detection

Errors that GRADES detects are broken down into verb-related errors and noun-related errors. In this section, the rules for seeking each GRADES error (including potential errors) will be discussed. Each rule will be accompanied by examples marked ungrammatical with * put at the front.

2.1 Verb-related errors

2.1.1 Sequence disorder of the verbal elements

(2-1-1a) Auxiliary verbs

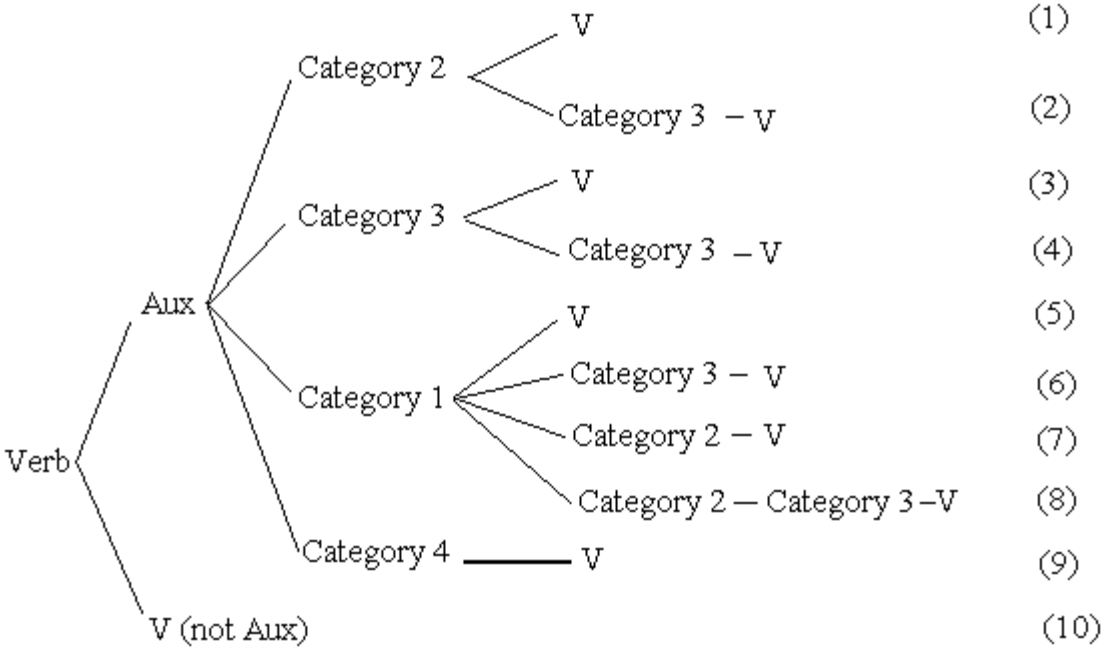
At this point in time, GRADES has already set up the verb element list, but the type of each element has not been identified yet. Thus, the first test to be performed is to check what type of verb element is used in the verb element list. If there are auxiliary verbs, they are checked to see if they are grammatically ordered. English auxiliary verbs can be categorized into the following four types.

Category 1….. **modals** (e.g. "can", "will", "should", etc),
Category 2….. **have** used in the present perfect tense like "I <u>have read</u> this book" or "He <u>has known</u> Sara for two years".
Category 3….. **be** used in the progressive aspect like "I <u>am cooking</u> supper the passive structure like "My house <u>was broken</u> into".
Category 4….. **do** used for emphasis or used in questions or negative sentences

There are two important grammatical rules related with auxiliary verbs. Those rules are:

a. Coexistence
The Categories 1, 2, 3 auxiliary verbs can co-exist, while the Category 4 auxiliary verb cannot co-exist with an auxiliary verb of any other category.
b. Precedence rule among Category 1, 2, 3
The **Category 1** auxiliary verb is followed by the **Category 2** auxiliary verb followed by the **Category 3** auxiliary verb.

Therefore, the verbal pattern of English can be determined out of the following ten possible patterns by checking the category of each auxiliary verb linearly.



Examples
Type (1) He has enjoyed the party.
Type (2) He has been teaching for many years.
Type (3) My mother is cooking now.
Type (4) I am being helped.
Type (5) I can play the viola.
Type (6) She will be getting better soon.
Type (7) I could have done it.
Type (8) He should have been studying then.
Type (9) I did not do it./ I do know the man.
Type (10) Tom goes for a walk every day.

Figure 5: Classification of the word order of the auxiliary verbs

Theoretically, in fact, there should be 3 more types in this list. They are:

Category 2 – Category3 – Category 3 – V
ex. How long *has* the research *been being done*?
Category 1 – Category 3 – Category 3 – V
ex. You will *be being told* in the near future.
Category 1 – Category 2 – Category 3 – Category 3 – V
ex. By next Christmas, that bridge *will have been being* built for three years.

According to Swan (1980), native speakers usually tend to avoid using "be being" and "been being", and those three types of sequences are very uncommon. Therefore, these patterns are excluded from the verb pattern checking process in GRADES.

The process of identifying the verbal pattern employs the same idea as top-down parsing. GRADES checks the order of auxiliary verbs and the location of a main verb based on the expected sequence after a certain auxiliary verb. For instance, GRADES analyzes the following verbal element sequence:

may have been having

GRADES finds "may" first. This is a Category 1 auxiliary verb, thus the system expects a Category 2 auxiliary verb (the form of "have") or a Category 3 auxiliary verb (the form of "be") or a verb to come next. Then, the system finds the word "have". The problem of "have" is that this word can be a verb (e.g. I <u>have</u> a cat) or an auxiliary verb (e.g. I <u>have</u> read that book). In order to determine the word's part of speech, GRADES checks if the word is placed at the end of the verb element list. If it is, the word has to be interpreted as a verb. If not, it has to be an auxiliary verb. Since "have" is not the last element of the list, it is concluded that the word "have" is an auxiliary verb, and this category meets the sequence restriction of "may". Now, GRADES focuses on what follows the auxiliary verb "have". This word takes either Category 3 (the form of "be") auxiliary verb or a verb. GRADES then finds "been", the form of "be". Like "have", "been" can be a verb (e.g. I have <u>been</u> there) or an auxiliary verb (e.g. I have <u>been</u> waiting for you). Therefore, as with determining the part of speech of "have", GRADES checks if "been" is located at the end of the list. Because it is not the last element in the list, "been" has to be interpreted as an auxiliary verb, which meets the sequence restriction

of “have”. In this case, only a verb can follow “been”. GRADES finds “having”, which can be a verb, and checks that this is the last element. Finally, GRADES concludes that this verbal sequence is Type (8) and that no grammatical errors are made in this phrase. When the sequence pattern is recognized as in this example, the phrase is given the verbal sequence number that corresponds to one of the types indicated in the figure 5 above. The number is later used when the verb form testing is performed later.

If GRADES finds some verbal element that does not meet the sequence restriction during the above process, a word order error is detected. For instance, suppose that the following phrase:

1. *I am having had fun.

is analyzed. The auxiliary verb “am” (Category 3) can only takes the Category 2 auxiliary verb (the form of “be”) or the verb. Since the next word having is either Category 3 auxiliary verb or the verb, the word “having” has to be interpreted as the verb to make this phrase correct. However, the word is not the last word of the verbal element list. There is another extra verbal element present. GRADES checks the verb sub-structure of “have”, which is that the verb may not be followed directly by another verb. Therefore, GRADES identifies “had” as Category 2 auxiliary verb. Since the verb having is followed by the Category 2 auxiliary verb, this is the violation of the sequence restriction. GRADES diagnoses the sequence of example 1 as a phrase out of order.

As in this example, when a sequence disorder error is detected, the verb phrase is not given any verbal pattern number, and further testing for verb-related errors will not be pursued.

(2-1-1b) Sub-structure of the main verb

Some of English verbs allow a second verb to follow, such as “want to travel” or “quit smoking”. In most cases, the form of the second verb has to be in the -ing form (e.g. swimming) or the infinitive with “to” (e.g. to swim). This structure is referred to as the “sub-structure” of the main verb in this report. What is important with the sub-structure of the main verb is that the second verb may show up on the verbal element list. This is the case when the verb is in the -ing form or infinitive without “to” as in the second and fourth example above. This is because GRADES picks any verbal elements to create the verbal list unless the verbal elements are separated by an inserted phrase or an adverbial phrase. Therefore, GRADES has to check the verb form of the second verb if there is any.

When GRADES finds out in the verb element sequence check that there is the second verb present in the verb element list, it checks if the first verb actually can take the second verb. If that is grammatical, the verb phrase is given the verb pattern number for the verb phrase with the main verb followed by the second verb. In other words, “have liked” and “have liked walking” are assigned two different verb pattern number.

2.1.2 Incorrect verb form

After the verb sequence pattern has been identified, the next step is to check if each verb is used in the correct form. The following are examples of an incorrect verb form:

1. *I have find a good present.
2. *I am have a good time.
3. *He may having lunch.

Based on the pattern number assigned as the result of the verb element sequence check, the elements in the verbal list are checked if the (auxiliary) verb meets the form restriction. For instance, in example 1 above, the verb element sequence check reveals that the verb phrase "have find" has the sequence of the Category 3 auxiliary verb followed by the main verb. Based on this analysis, GRADES starts with going to the first auxiliary verb "have" and checks if the second verb is in the past participle form. In this case, "find" is not in the form as the syntax rule says. Therefore, GRADES finds the violation and diagnoses this as an incorrect verb form.

Checking for an incorrect verb form also extends to the second verb after the main verb. Based on Hornby (1976), Frank (1972) and Swan (1980), the English verbs have four types of sub-structures.

1. Verbs that only take the -ing form
   ex. She enjoys swimming.
2. Verbs that only take infinitive with "to"
   ex. I plan to buy a new car.
3. Verbs that take either one of the verb form with little semantic difference
   ex. I love to read. / I love reading.
4. Verbs that take either one of the verb form with significant semantic difference
   ex. I remember talking to you. / I remember to talk to you.

Examples of errors of incorrect sub-structure form are:

1. *She enjoys *to swim*.
2. *She enjoys *swim*.
3. *I plan *buying* a new car.
4. *I plan *buy* a new car.

Whether there is the second verb is present is encoded in the verb element sequence pattern number if the second verb is in the -ing form (such as "enjoy cooking") and appears in the verb element list. If GRADES is given the number, it checks the sub-structure type of the main verb. If it takes the second structure, GRADES gets the information on the required form on the second verb. Next, GRADES simply goes to the next verb after the main verb in the verb element list and gets the information of its form. If GRADES finds that the required form and the form actually used in the input do not match, it concludes that there is an incorrect verb form error.

If the second verb takes the infinitive form with "to" (such as "try to do it"), it does not show in the verb element. GRADES must go to the input list, which is the original input sentence, to find whether the second verb is present. Therefore, every time the verb elements sequence number indicates that no second verb is found in the verb element list. GRADES goes to the input list after reaching the last element in the verb element list and checks on the two words immediately after the main verb in the input list. If they are "to" and a verb, GRADES gets information on the substructure of the main verb and then, the information on the form of the second verb. If the forms do not match, GRADES concludes that an incorrect verb form error exists.

2.1.3 Subject-verb / Subject-auxiliary verb disagreement

In an English sentence, the (auxiliary) verb has to agree with the subject in terms of person and number. The following are examples of the violation on subject-verb or subject-auxiliary verb agreement.

1. *The *man have* (main verb) a nice car.
2. *My *friends speaks* (main verb) Spanish.
3. *My *brother have* (auxiliary verb) won the first prize.

The first verb element in the VP, auxiliary verb or verb, is the only one that has to agree with the subject, but there are exceptions among the auxiliary verbs. A subject-auxiliary verb agreement does not apply to the modal auxiliary verbs (such as "can", "may" or "will"). Therefore, more generally speaking, one can say that in English, the first verb element of the VP has to agree with the subject in terms of person and number unless the first verb element is a modal auxiliary verb (Category 1).

In order to check the agreement, GRADES compares the person and number features of both the subject and the first verb element. First, GRADES finds the subject in the input list. Because the subject is already specified at the time of input, GRADES simply looks for the information from the user to identify the subject. GRADES checks if the subject is either a noun or pronoun and then, gets its person and number information from the corresponding part of speech list. Next, GRADES goes directly to the first element of the verb element list. Because the type of the verbal element sequence is identified at this point, the part of speech of the first verbal element is also identified. GRADES gets the part of speech of the word and finds the word in the part of the speech list to access the person and number attribute. If the VP only consists of the main verb, a subject-verb disagreement check is performed. The verbal element list looks like:

Verb element list: run

GRADES looks for the number information on the word run in the verb list. If the VP contains auxiliary verbs, a subject-auxiliary verb disagreement is performed here. The list looks like:

Verb element list: has been running

GRADES looks for the person and number information on the word has in the auxiliary verb list.

If the person and number information of the (auxiliary) verb does not match the person and number information of the subject, the disagreement error is detected.

2.1.4 Lack of the auxiliary verb "do" for the negation

In English, when the sentence is negated, the auxiliary verb "do" has to be used if the VP only consists of the main verb that is not the verb "be". The sentence is ungrammatical if the auxiliary verb "do" is absent. The following is an example of this error type.

1. *He not work here.

In the process of identifying the verb element sequence, this erroneous structure is categorized into Type (10) in figure 2 as successfully as the grammatical sentence "He works here".

GRADES can detect this error in the following way. GRADES checks the VP categorized into Type (10) if it is affirmative or negative by checking the value of the Negation flag. If the sentence is affirmative, a subject-verb disagreement check is performed. If the sentence is negative, a subject-verb disagreement check is not performed. Instead, GRADES checks if the main verb is the verb "be" or not. If it is not (that is, a regular verb), GRADES concludes that the auxiliary verb "do" is absent.

2.1.5 Incorrect use of the auxiliary verb "be" with an intransitive verb

Even though the verb is in the correct form, the passive voice structure is ungrammatical if it is used with an intransitive verb. Examples of this error type are:

1. *That *is* happened to me.
2. *She *is* arrived.

These sentences become grammatical if the auxiliary verb is "has" instead of "is".

During the verb element sequence check, these structures are categorized into Type (3) in figure 2. Then, during the verb form check, it is diagnosed as a usage error, but the cause is not specified yet. During the test for this type of error, GRADES checks the VP categorized into this Type (3) if the main verb can be a transitive verb. If it is only used intransitively, GRADES diagnoses the "be + past participle" structure as an incorrect use of the auxiliary verb "be".

2.1.6 Incorrect use of the passive voice

This type of error occurs when the verb is used in the passive voice structure and yet the verb has the object after it. For instance,

1. *The girl *was damaged* her car.
2. *He may *be stolen* money.

Because the sequence of the verb elements are correct, these VPs are identified as Type (3) in figure 2. No errors are found in the verb form check, either. The same type of error also can occur in Type(2), Type(6) and Type(8), where the passive voice is used.

GRADES identifies the error in the following way. It checks the VP categorized into those verb patterns if the verb can be transitive and the object noun is present. GRADES searches for the verb in the verb list to get the information on transitivity of the verb. GRADES checks the value of the VerbHasObject flag to find if there is the object. If GRADES finds that the verb is transitive and has the object noun, it diagnoses the use of the passive voice structure as an error.

2.1.7 Lack of the auxiliary verb "be" for the passive voice

This error occurs when the verb "be" is absent and the noun, which is semantically the object of the verb in the active meaning, becomes the subject of the sentence. In other words, the auxiliary verb "be" is absent when it is syntactically needed. Examples of this error type are:

1. *My car __ stolen.
2. *I __ asked to go with him.

These two examples are both categorized into Type (10) in figure 2. With example 1, the use of past participle is not correct, thus the incorrect verb form check identifies a verb form error. With example 2, however, a verb form error is not even detected because the verb "asked" can be the past tense form as well as the past participle form. The same type of error can be found in Type(1), Type (5), and Type (7) as well.

GRADES detects this error in the following way. It checks the VPs categorized into those types on two pieces of syntactic information: the verb form and the presence of the object. If the verb form is the past participle and also, the object is absent, GRADES checks on the part of speech of the verb. If it only functions as a transitive verb, GRADES diagnoses this structure as lacking the auxiliary verb "be". If the verb can be an intransitive or transitive verb, GRADES diagnoses the structure as a potential error.

2.1.8 Lack of the main verb

The error of this type occurs when the sentence lacks the verb after "did not" or "do not" or in the question starting with "do", "did" or "does". For instance,

1. *I did not fever.
2. *Did you fever?

However, since the word "do" / "did" / "does" is the only element in the verb element list, it has to be interpreted as a verb instead of an auxiliary verb. As a result, the VP of this type of structure is categorized into Type (10).

GRADES has to distinguish such structures and sentences where "do" / "did" / "does" is actually used as a verb. In order to do so, GRADES checks the VP of Type (10) on Negation and the DoQuestion flag. If the Negation flag is set to 1, GRADES concludes that "do"/ "did"/ "does" has to be an auxiliary verb syntactically and detects an error of the absence of the main verb. Likewise, if the DoQuestion is set to 1, GRADES concludes that "do" / "did" / "does" has to be also an auxiliary verb and the sentence lacks the main verb.

The following three rules are rules for seeking *potential errors*. A potential error here is an error type that may be caused by one of the types described here. The errors, in these cases, are either incorrect choice or deletion of an auxiliary verb. Because there are two possible causes of these errors, GRADES is not be able to differentiate which of the two is an actual cause.

2.1.9 Incorrect use of the auxiliary verb be for the negation

This type of error may be the case when the Category 2 auxiliary verb (the form of "be") is used with a regular verb in a negative sentence. For instance,

1. *He *is not think* so.

The auxiliary verb "is" may be mistakenly chosen instead of the auxiliary verb "does".

After categorizing the VP into Type (3), GRADES detects the verb form error on the verb "think" after the verb "be". Separately from this verb form testing, GRADES checks the VP of Type (3) on the verb form and the presence of the negative particle "not". If the verb form is in the present tense form and the Negation flag is set to 1, GRADES gives a warning message that the word "is" may be an incorrect choice for the intended meaning.

2.1.10 Lack of the auxiliary verb "be" for the progressive tense

Lack of the form of "be" may be the cause when the verb is used in the -ing form by itself where the other forms have to be used. In other words, the verb occurs in the -ing form in the first position of the verb element list, or immediately after the form of "have" or the modal auxiliary verb like "may" or "will". The following shows this type of errors:

1. *I *have thinking* about you.
2. *I *thinking* about you.
3. *I *will thinking* about you.

The above examples 1, 2, and 3 are categorized into Type (10), Type (1), and Type (5) respectively, and their main verbs are diagnosed as an incorrect verb form. GRADES checks the VP of these patterns on their verb form. If the verb form is the present participle, GRADES provides a warning message to suggest that the form of "be" may be forgotten.

2.1.11 Lack of the auxiliary verb "have" for the present perfect tense

The lack of the auxiliary verb "have" may occur when the verb is used in the past participle form by itself where the other forms should be used. In other words, the verb of the past participle form occurs in the first position of the verb element list or immediately after the modal auxiliary verb. For instance,

1. *She *may found* the dog.
2. *She *gone* there before.

Examples 1 and 2 are categorized into Type (5) and Type (10) respectively, and their main verbs are diagnosed as an incorrect verb form. Next, GRADES first checks the VP of these types on the verb form again and the part of speech of the main verb. If the verb takes the past participle and it is an intransitive verb, GRADES provides a warning that the verb may lack the auxiliary verb "have".

If the verb takes the past participle and it can be a transitive verb, GRADES next checks the value of the VerbHasObject flag. If it is set to 1, GRADES also provides the same warning.

2.2 Noun-related errors

2.2.1 Number disagreement in the NP

If a countable noun is modified by a word that indicates more than one item (ex, numbers larger than one, "many", "some", "those" or "these"), the noun has to take the plural form. The following examples show the violation on noun number agreement.

1. *I have five sister.
2. *Many child came to the concert.

In the noun number agreement check, instead of searching for a determiner or adjective, which is the case in parsing for the NP, GRADES first looks for a noun in the sentence. As

soon as it finds one, the noun's number feature is checked. Next, GRADES tries to find if the noun is immediately preceded by a modifier (an adjective or a determiner). If so, GRADES checks the number feature of the modifier. Adjectives and determiners that require the plural form on the noun are mainly numbers and quantifiers, such as "many" or "some", in English.

In example 1, GRADES finds the noun "sister". It is in the singular form. Next, GRADES goes back to the previous word in the input list to check if the noun has a modifier. GRADES finds "five", which is a determiner and requires the plural form on the modified noun. GRADES compares the number requirement and the number actually expressed by the noun. Because they do not match in this case, GRADES diagnoses "five sister" as a number disagreement error. As explained before, this process continues until no more modifiers are found. One advantage of this approach over parsing is efficiency. Unlike parsing, because the noun is first found, the number agreement between the modifier and the noun can be checked while the structure of premodifiers is analyzed.

As the result of this testing, the AdjPresent carries 0 because there is no adjective found and the DetPresent 1 because the determiner "five" is found.

2.2.2 Lack of determiner

This type of error occurs when there is no determiner used when a countable singular noun is the noun head under the condition that the noun refers to something unspecific. The following examples show the lack of a determiner:

1. *I have ___ sister.
2. *I bought ___ nice red car.

By the time of launching the test for the lack of determiner, GRADES has finished the noun disagreement test and knows if the modifiers of the noun contain a determiner in the 1st premodifier position. GRADES checks the DetPresent flag. In the both cases above, the flag is set to 0 because there is no determiner used. If the flag is set to 0 like here, GRADES checks the countability of the noun. If it is countable, GRADES diagnoses the absence of a determiner as an error.

2.2.3 Extra determiner

This type of error occurs when two determiners are used to modify one noun. For instance,

1. *We like his this poem.
2. *That her camera is expensive.

The process of checking the number agreement stops when the GRADES encounters a determiner. In the above examples, the determiners "this" of 1 and "her" of 2 terminate the process of testing a number disagreement error.

In this test for extra determiner, GRADES first checks the DetPresent flag to find out if the modifier phrase has a determiner. If it is set to 1, GRADES checks on the part of speech of a word immediately before the determiner. If the word is also a determiner, GRADES finds the violation that more than one determiner is used together as premodifiers for the same noun and diagnoses multiple determiners next to each other as an error.

2.2.4 Modifiers out of order

This error occurs when an adjective is used before a determiner. Examples are:

1. *She likes red these shoes.
2. *The man drives old his car.

The test for the type of this error is also applied to the modifier phrase that has a determiner. If the DetPresent flag is set to 1, GRADES checks on the part of speech of a word immediately before the determiner. If the word is an adjective, GRADES diagnoses the sequence of two modifiers as a word order error.

RUN-TRACES

This section will present actual run-traces of GRADES on a number of examples sentences. The input sentence and the identified error are printed in boldface for better readability. Note that the run-traces do not show the user identified subject although this is required in the input.

A. Examples with verb-related errors

(1) He have enjoyed reading books.

INPUT SENTENCE ->he have enjoyed reading books .

Step 1: Checking verb elements out of order
Done. 1 Aux V, 2 Verbs in order

Step 2: Checking incorrect verb forms
present perfect form Done. OK
second verb Done. OK

Step 3: Checking Subject-Aux V disagreement Done.
Error: Subject(he)- Aux V(have) Number disagreement

Step 4: Checking lack of auxiliary verb (be) for passive Done. OK
Passive (be) not missing.

Step 5: Checking noun (books) number disagreement Done. Number in agreement

(2) The boy plan buy a car.

INPUT SENTENCE ->the boy plan buy a car .

Step 1: Checking verb elements out of order
Done. 2 Verbs in order

Step 2: Checking incorrect verb forms

verb form of main verb and no auxiliary verbs Done. OK
second verb Done.
Error: 'to' missing before 2nd Verb(buy)

Step 3: Checking Subject-Verb disagreement Done. Done.
Error: Subject(boy)- Main V(plan) Number disagreement

Step 4: Checking noun (boy) number disagreement Done. Number in agreement

Step 5: Checking modifiers out of order Done. OK Modifier(s) in order

Step 6: Checking noun (car) number disagreement Done. Number in agreement

Step 7: Checking extra determiner Done. OK No extra determiner found

(3) This is happened.

INPUT SENTENCE ->this is happened .

Step 1: Checking verb elements out of order
Done. 1 Aux V, 1 Verb in order

Step 2: Checking incorrect verb forms
progressive/passive form Done. Usage error, later checked

Step 3: Checking Subject-Aux V disagreement
Done. Subject-Aux V in agreement

Step 4: Checking misuse of aux verb (be) with intransitive verb Done.
Error: Auxiliary verb(is) incorrect choice. It should be (has)

(4) The book written.

INPUT SENTENCE ->the book written .

Step 1: Checking verb elements out of order
Done. 1 Verb in order

Step 2: Checking incorrect verb forms
verb form of main verb and no auxiliary verbs Done.
Error: Main verb(written) incorrect form

Step 3: Checking lack of auxiliary verb (be) for passive Done.
Error: Main verb(written) must be the passive voice with auxiliary verb(is/was) for the passive

Step 4: Checking noun (book) number disagreement Done. Number in agreement

Step 5: Checking modifiers out of order Done. OK Modifier(s) in order

(5) He has great ideas and write many books.

INPUT SENTENCE ->he has great ideas and write many books .

Step 1: Checking verb elements out of order
Done. 1 Verb in order

Step 2: Checking Subject-Verb disagreement Done. Subject-Verb in agreement.

Step 3: Checking lack of auxiliary verb (be) for passive Done. OK
Passive (be) not missing.

Step 4: Checking verb elements out of order
Done. 1 Verb in order

Step 5: Checking Subject-Verb disagreement Done. Done.
Error: Subject(he)- Main V(write) Number disagreement

Step 6: Checking lack of auxiliary verb (be) for passive Done. OK
Passive (be) not missing.

Step 7: Checking noun (ideas) number disagreement Done. Number in agreement

Step 8: Checking noun (books) number disagreement Done. Number in agreement

Step 9: Checking extra determiner Done. OK No extra determiner found

Step 10: Checking modifiers out of order Done. OK Modifier(s) in order

B. Examples with noun-related errors

(1) I love his that car.

INPUT SENTENCE ->I love this his car .

Step 1: Checking verb elements out of order
Done. 1 Verb in order

Step 2: Checking incorrect verb forms
verb form of main verb and no auxiliary verbs Done. OK

Step 3: Checking lack of auxiliary verb (be) for passive Done. OK
Passive (be) not missing.

Step 4: Checking noun (car) number disagreement Done. Number in agreement

Step 5: Checking extra determiner Done.
Error: 2 determiners(his and this) cannot be used together

Step 6: Checking modifiers out of order Done. OK Modifier(s) in order

(2) I have very bright student

INPUT SENTENCE ->I have very bright student .

Step 1: Checking verb elements out of order
Done. 1 Verb in order

Step 2: Checking incorrect verb forms
verb form of main verb and no auxiliary verbs Done. OK

Step 3: Checking Subject-Verb disagreement Done. Subject-Verb in agreement.

Step 4: Checking potential error of aux V (have) deletion Done.
No auxiliary verb (have) necessary

Step 5: Checking noun (student) number disagreement Done. Number in agreement

Step 6: Checking lack of determiner Done.
Error: Determiner necessary before adverb(very)

(3) I have new three great airplane.

INPUT SENTENCE ->I have new three great airplane .

Step 1: Checking verb elements out of order
Done. 1 Verb in order

Step 2: Checking incorrect verb forms
verb form of main verb and no auxiliary verbs Done. OK

Step 3: Checking lack of auxiliary verb (be) for passive Done. OK
Passive (be) not missing.

Step 4: Checking noun (airplane) number disagreement Done.
Error: Determiner(three)- Noun(airplane) Number disagreement

Step 5: Checking extra determiner Done. OK No extra determiner found

Step 6: Checking modifiers out of order Done.
Error: Adjective(new) and Determiner (three) out of order

C. Example with both verb-related errors and noun-related errors

Many girl may have be enjoy to read my these book.

INPUT SENTENCE ->many girl may have be enjoy to read my these book .

Step 1: Checking verb elements out of order
Done. 3 Aux Vs, 1 Verb in order

Step 2: Checking incorrect verb forms
verb form after modal Done. OK
present perfect form Done.
Error: Auxiliary verb(be)incorrect form
progressive/passive form Done.
Error: Main verb(enjoy)incorrect form after verb-be

Step 3: Checking Subject-Aux V disagreement
Done. Subject-Aux V in agreement

Step 4: Checking lack of auxiliary verb (be) for passive Done. OK
Passive (be) not missing.

Step 5: Checking noun (girl) number disagreement Done.
Error: Determiner(many)- Noun(girl) Number disagreement

Step 6: Checking modifiers out of order Done. OK Modifier(s) in order

Step 7: Checking noun (book) number disagreement Done.
Error: Determiner(these)- Noun(book) Number disagreement

Step 8: Checking extra determiner Done.
Error: 2 determiners(these and my) cannot be used together

Step 9: Checking modifiers out of order Done. OK Modifier(s) in order

FUTURE WORK

In order to make GRADES more complete and effective as an error detection system, more improvements should be made at least in the following four areas.

A. Input requirement

At this point, GRADES requires that the user indicate the subject word at the time of input, but some mechanism needs to be implemented that finds the subject without interactions with the user.

B. Input restrictions

GRADES needs to be able to process not only simple sentences but also complex and compound sentences. Examples of these sentences are:

Relative clauses with/without a relative pronoun

(1) I know the man who is good at programming.
(2) I bought a gift (that) my mother wanted.

Adverbial clauses preceded by a sub-ordinate conjunction, such as “if”, “when”, “while”, etc.

(3) If it rains tomorrow, the picnic will be cancelled.
(4) The picnic will be cancelled if it rains tomorrow.

Noun clauses preceded by a sub-ordinate conjunction, such as “that”, “if” or “whether”.

(5) I said that he should take a break.

Multiple independent sentences conjoined by a coordinate conjunction, such as “and”, “but”, “or”, “so”, and “for”

(6) Mike likes beef, and Helen likes fish.

In order to be able to process the sentence structures listed above, GRADES needs to do structural analysis so that a sentence, containing multiple clauses, can be broken into simple sentences. With adverbial and noun clauses, such as examples (3), (4), and (5), at lease two tests would be implemented: one test to recognize conjunctions and a further test to find the end of the adverbial clause if the adverbial clause is placed before the main clause as in example (3). The end of an adverbial clause might be able to be found by searching for a comma although there is no guarantee that a comma is used. The end of an adverbial clause would also be found by trying to find a (auxiliary) verb and then, the closest NP to it, which is a strong indication of the presence of a sentence. In this case, however, the NP that occurs after a verb and has the noun head in the object case, such as “him” in “I heard him sing”, has to be excluded.

With multiple independent sentences conjoined by a coordinate conjunctions, such as example (6), two tests would be also implemented: one test to recognize a coordinate conjunction and another to check if the conjunction is followed by the NP that has the noun head in the subject case and V, that is, a sentence.

Tests for relative clauses are probably the most difficult to implement. One of the most crucial operations to be performed with relative clauses is to find the beginning and end of the clause when it modifies the subject of the sentence because the verb of the sentence has to be found to test a subject-verb / subject-auxiliary verb disagreement. Therefore, the best strategy to delimiting two clauses here would be to find the verb that should match up with the subject by eliminating all possible NP (as the subject)-V combinations instead of trying to find the last word of the relative clause. If a relative pronoun is used, that word is the beginning of the relative clause. First, an antecedent, the noun before the relative pronoun, would be stored. Next, the first verb after the relative pronoun would be sought. Once the verb is found, it would be checked if there is the NP between the verb and the relative pronoun. If there is, the relative pronoun functions as the object of the relative clause. If there is none, the relative pronoun functions as the subject of the relative clause. The relative pronoun of this type should be

included in the next NP searching operation. Now, the system would search back for the closest NP to the verb. If the NP that is found is not the same as the antecedent, it is still within the relative clause. Then, the second verb is sought. When the second verb is found, seeking the closest NP to the second verb that is not checked before would take place. The same procedure would be repeated until the antecedent is reached.

If there is no relative pronoun used, extra process for identifying the presence of a relative clause is necessary in addition to the above operations. Every time the NP is found, the sentence would be checked if the noun head is immediately followed by a relative clause or another NP. If there is another NP found, it is likely that this second NP is the beginning of the relative clause. In order to check if the second NP forms a sentence, a series test would be implemented. First, it would be checked if the verb is present after the second NP. If it is, the verb would be checked if it lacks an object or one of the objects. If it does, it would be highly probable that the second NP is the beginning of a relative clause.

C. Additional rules used for error detection

GRADES can diagnose seventeen error categories, but these are only part of the grammatical errors that can be found among non-native speakers' errors. The following is a list of errors that are to be implemented in the future. See details in the Appendix.

- Incorrect choice of pronoun
- Tense mismatch
- Incorrect modification by an adverb
- Incorrect modification by an adjective
- Incorrect form of a pronoun
- Incorrect use of a preposition
- Lack of element (articles, prepositions and subject)
- Extra element (articles, prepositions and subject)
- Incorrect word order

D. Enlarging the lexicon

If input can be complex and compound sentences mentioned in B, another goal has to be to have a larger lexicon to reflect a greater portion of the English sentence.

CONCLUSION

Grammar checkers are potentially a great tool to provide individual tutoring assistance for adult second language learners. Unfortunately, however, grammar checkers are not designed to detect non-native speakers' errors. Some common errors, which are quite obvious to native speakers, are not detected, thus no explanation of the cause is given. Even misdiagnosis occurs.

GRADES, reported on in this paper, is built using diagnostic strategies instead of natural language processing, used typically in grammar checkers. GRADES also takes into considerations of types of errors that non-native speakers commonly make. In grammar checkers, errors are detected by identifying the syntactic structures of a sentence and creating a parse tree. Based on the parse tree, syntactic rules are checked if they are met in the particular sentence. On the other hand, GRADES is a rule-based diagnostic expert system. Using a set of the relevant rules for a certain type of error, GRADES narrows down a specific category for identifying the type of error in the form of hierarchical classification.

GRADES diagnoses the following seventeen types of errors: thirteen verb-related errors and four noun-related errors.

Verb-related errors

- Subject-verb disagreement
- Subject-auxiliary verb disagreement
- Verbal elements out of order (auxiliary verbs and main verb)
- Incorrect verb form of the verbal elements (verb form requirement by an auxiliary verb immediately before)
- Incorrect verbal sub-structure of the verb (verb form when the verb is used right after the main verb, such as "avoid driving" vs. *"avoid to drive")
- Lack of the auxiliary verb "do" for the negation
- Incorrect choice of the auxiliary verb "be" with an intransitive verb
- Incorrect use of passive voice (the presence of verb "be" when not needed)
- Lack of the auxiliary verb "be" for the passive voice (the absence of verb "be" when needed)
- Lack of the main verb

"potential" errors

- Incorrect choice of the auxiliary verb "be" in the negation
- Lack of the auxiliary verb "be" for the progressive tense
- Lack of the auxiliary verb "have" for the present perfect

Noun-related errors

- Noun number disagreement in NP
- Lack of a determiner
- Extra determiner
- Modifiers out of order

Error detection mechanisms have always been part of natural language processing. This paper has intended to report another possibility: a diagnostic expert system. When non-native speakers' errors are diagnosed, there are so many varieties of possible errors because they do not always successfully apply grammatical rules. Using very specific rules and focusing on common error types, a grammar checker as an expert system may possibly detect non-native speakers' errors more accurately than regular grammar checkers, which makes a grammar checker as an expert system more effective tutoring tool for non-native speakers of English.

APPENDIX A

The following are other types of errors to be implemented. They are accompanied with error examples.

1. Disagreement
   - Incorrect choice of pronoun
     A pronoun referring back to the characters previously introduced does not agree with the character's number and person.

     My husband and I asked the front desk to give *me* (meaning *us*) wake-up call.

   - Tense mismatch
     The present tense is used when the past tense has to be used.

     I *do* not (meaning *did* not) understand.
     When she *is* fifteen, she entered a local high school in Edinburg.

2. Incorrect modification
   - Adverb
     An adverb is used in the following syntactic environments: (a) modifying a noun in front, (b) after a linking verb, such as *be*, *look* or *sound*.

     We ate meals of *differently* kinds of fish.
     She looks *differently*.

   - Adjective
     An adjective is used to modify a verb. Also, *this* or *that* is used with a possessive pronoun, such as *his* or *my*.

     He talks *clear*.
     This my book is very important.

3. Incorrect use of element
   - Pronoun
     The object case is used in the subject position.

     My mother and *me* went shopping together.

   - Preposition
     The choice of a preposition is incorrect, syntactically or semantically.

     Thank you very much *to* your time.
     You have to send this *until* March.

4. Lack of element

- Article

  No definite article is used when it is necessary, syntactically or semantically.

  ___ First thing for you to do is to look for a job.
  I know ___ man.(when a specific man is already introduced)

- Preposition

  No preposition is used when it is necessary, syntactically or semantically.

  My son said __ her, “Sit down”.
  When I was __ the 2nd grade…

- Subject

  No subject is present.

  I had walked to the beach many times since __ met the fisherman…

5. Extra element

- Article

  An article is used when it is not necessary.

  He sent to *the* high school in McAllen.

- Preposition

  A preposition is used when it is not necessary.

  only once *in* the month…
  in last week…

- Subject/object in a relative clause

  The antecedent of the relative clause is the subject/object of the relative clause, but the subject/object appears in the clause.

  a family that *it* has many boys

6. Incorrect word order

- Pronoun

  A pronoun follows some verb phrases, such as *put on* or *throw out*.

  I threw out it.

- Adverb placement

  An adverb (expression) is placed between the verb and the object of the sentence.

  I know *very well* him.

Chris sang *carefully* the song.

- Embedded interrogative clauses
  Embedded questions, which follow such clauses as *I wonder* or *I asked*, have the word order of independent questions.

  I wonder *how did he do it*.
  I did not know *why did she say that*.

APPENDIX B

The following is a list of the words in the current lexicon of GRADES. Various lexical information stored in each word is shown in the parentheses following the word.

Auxiliary verbs

Lexical information (Category of the auxiliary verb, Form, Subject type to be used)

abbr.
Form original: OR, present tense: PRE, past tense: PAST, past participle: PASTP, present participle: PREP
Types of subject to be used with the word
any noun: ANYS, 3rd person singular: 3PS, other than 3rd person singular noun: N3PS, no subject: NOS

would (modal, PAST, ANYS)
should (modal, PAST, ANYS)
could (modal, PAST, ANYS)
might (modal, PAST, ANYS)
will (modal, PRE, ANYS)
may (modal, PRE, ANYS)
can (modal, PRE, ANYS)
have (have-type, OR and PRE, N3PS)
had (have-type, PAST and PASTP, ANYS)
has (have-type, PRE, 3PS)
be (be-type, OR, NOS)
been (be-type, PASTP, NOS)
being (be-type, PREP, NOS)
am (be-type, PRE, with "I")
is (be-type, PRE, 3PS)
was (be-type, PAST, with "I" or 3PS)
are (be-type, PRE, with "you" and a plural subject)
were (be-type, PAST, with "you" and a plural subject)
do (do-type, OR and PRE, N3PS)
does (do-type, PRE, 3PS)

Verbs

Lexical information (Category of the verb, Form, Verbal sub-structure, Subject type to be used)

abbr.
Category intransitive: INT, transitive: TRAN, linking: LINK
Form original: OR, present tense: PRE, past tense: PAST, past participle: PASTP, present participle: PREP

Type of the sub-structure

takes to + verb(original form): INF
takes verb (–ing form): GER
takes to + verb(original form) or verb (–ing form): INF/GER
takes to + verb(original form) or verb (–ing form) with a semantic difference: INF/GER(D)
takes no second verb: NO2V

Types of subject to be used with the word

any noun: ANYS, 3rd person singular: 3PS, other than 3rd person singular noun: N3PS,
no subject: NOS

walk (INT, OR and PRE, INF, N3PS)
walked (INT, PAST and PASTP, INF, ANYS)
walking (INT, PREP, INF, NOS)
walks (INT, PRE, INF, 3PS)
be (LINK, OR, INF, NOS)
am (LINK, PRE, INF, "I")
is (LINK, PRE, INF, 3PS)
are (LINK, PRE, INF, "you" and a plural subject)
were (LINK, PAST, INF, "you" and a plural subject)
was (LINK, PAST, INF, 3PS)
been (LINK, PASTP, INF, NOS)
being (LINK, PREP, INF, NOS)
have (TRAN, PRE, INF, N3PS)
has (TRAN, PRE, INF, 3PS)
had (TRAN, PAST and PASTP, INF, ANYS)
having (TRAN, PREP, INF, NOS)
write (TRAN, OR and PRE, NO2V, N3PS)
writes (TRAN, PRE, NO2V, 3PS)
wrote (TRAN, PAST, NO2V, ANYS)
written (TRAN, PASTP, NO2V, NOS)
writing (TRAN, PREP, NO2V, NOS)
read (TRAN, OR / PRE / PAST / PASTP, NO2V, ANYS)
reads (TRAN, PRE, NO2V, 3PS)
reading (TRAN, PREP, NO2V, NOS)
enjoy (TRAN, PRE, GER, N3PS)
enjoys (TRAN, PRE, GER, 3PS)
enjoyed (TRAN, PAST and PASTP, INF, ANYS)
enjoying (TRAN, PREP, GER, NOS)
think (TRAN, OR and PRE, INF, N3PS)
thinks (TRAN, PRE, INF, 3PS)
thought (TRAN, PAST and PASTP, INF, ANYS)
thinking (TRAN, PREP, INF, NOS)
try (TRAN, OR and PRE, INF, N3PS)
tries (TRAN, PRE, INF, 3PS)
tried (TRAN, PAST and PASTP, INF, ANYS)
trying (TRAN, PREP, INF, NOS)

exchange (TRAN, OR and PRE, NO2V, N3PS)
exchanges (TRAN, PRE, NO2V, 3PS)
exchanged (TRAN, PAST and PASTP, NO2V, ANYS)
exchanging (TRAN, PREP, NO2V, NOS)
do (TRAN, PRE, NO2V, N3PS)
does (TRAN, PRE, NO2V, 3PS)
did (TRAN, PAST, NO2V, ANYS)
done (TRAN, PASTP, NO2V, NOS)
doing (TRAN, PREP, NO2V, NOS)
begin (TRAN, PRE, INF, N3PS)
began (TRAN, PAST, INF, ANYS)
begun (TRAN, PASTP, INF, NOS)
beginning (TRAN, PREP, INF, NOS)
begins (TRAN, PRE, INF, 3PS)
love (TRAN, OR and PRE, INF, N3PS)
loved (TRAN, PAST and PASTP , INF, ANYS)
loving (TRAN, PREP, INF, NOS)
loves (TRAN, PRE, INF, 3PS)
start (TRAN, OR and PRE, INF/GER, N3PS)
started (TRAN, PAST and PASTP, INF/GER, ANYS)
starting (TRAN, PREP, INF/GER, NOS)
starts (TRAN, PRE, INF/GER, 3PS)
intend (TRAN, OR and PRE, INF/GER, N3PS)
intended (TRAN, PAST and PASTP, INF/GER, ANYS)
intending (TRAN, PREP, INF/GER, NOS)
intends (TRAN, PRE, INF/GER, 3PS)
hate (TRAN, OR and PRE, INF/GER, N3PS)
hated (TRAN, PAST and PASTP, INF/GER, ANYS)
hating (TRAN, PREP, INF/GER, NOS)
hates (TRAN, PRE, INF/GER, 3PS)
choose (TRAN, OR and PRE, INF, N3PS)
chose (TRAN, PAST, INF, ANYS)
chosen (TRAN, PASTP, INF, NOS)
choosing (TRAN, PREP, INF, NOS)
chooses (TRAN, PRE, INF, 3PS)
decide (TRAN, OR and PRE, INF, N3PS)
decided (TRAN, PAST and PASTP, INF, ANYS)
deciding (TRAN, PREP, INF, NOS)
decides (TRAN, PRE, INF, 3PS)
hope (TRAN, OR and PRE, INF, N3PS)
hoped (TRAN, PAST and PASTP, INF, ANYS)
hoping (TRAN, PREP, INF, NOS)
hopes (TRAN, PRE, INF, 3PS)
plan (TRAN, OR and PRE, INF, N3PS)
planned (TRAN, PAST and PASTP, INF, ANYS)
planning (TRAN, PREP, INF, NOS)
plans (TRAN, PRE, INF, 3PS)
wish (TRAN, OR and PRE, INF, N3PS)
wished (TRAN, PAST and PASTP, INF, ANYS)

wishing (TRAN, PREP, INF, NOS)
wishes (TRAN, PRE, INF, 3PS)
happen (INT, OR and PRE, INF, N3PS)
happened (INT, PAST and PASTP, INF, ANYS)
happening (INT, PREP, INF, NOS)
happens (INT, PRE, INF, 3PS)
forget (TRAN, OR and PRE, INF/GER(D), N3PS)
forgot (TRAN, PAST, INF/GER(D), ANYS)
forgotten (TRAN, PASTP, INF/GER(D), NOS)
forgetting (TRAN, PREP, INF/GER(D), NOS)
forgets (TRAN, PRE, INF/GER(D), 3PS)
remember (TRAN, OR and PRE, INF/GER(D), N3PS)
remembered (TRAN, PAST and PASTP, INF/GER(D), ANYS)
remembering (TRAN, PREP, INF/GER(D), NOS)
remembers (TRAN, PRE, INF/GER(D), 3PS)
stop (TRAN, OR and PRE, INF/GER(D), N3PS)
stopped (TRAN, PAST and PASTP, INF/GER(D), ANYS)
stopping (TRAN, PREP, INF/GER(D), NOS)
stops (TRAN, PRE, INF/GER(D), 3PS)
want (TRAN, OR and PRE, INF/GER(D), N3PS)
wanted (TRAN, PAST and PASTP, INF/GER(D), ANYS)
wanting (TRAN, PREP, INF/GER(D), NOS)
wants (TRAN, PRE, INF/GER(D), 3PS)
finish (TRAN, OR and PRE, GER, N3PS)
finished (TRAN, PAST and PASTP, GER, ANYS)
finishing (TRAN, PREP, GER, NOS)
finishes (TRAN, PRE, GER, 3PS)
mind (TRAN, OR and PRE, GER, N3PS)
minded (TRAN, PAST and PASTP, GER, ANYS)
minding (TRAN, PREP, GER, NOS)
minds (TRAN, PRE, GER, 3PS)
miss (TRAN, OR and PRE, GER, N3PS)
missed (TRAN, PAST and PASTP, GER, ANYS)
missing (TRAN, PREP, GER, NOS)
misses (TRAN, PRE, GER, 3PS)
understand (TRAN, OR and PRE, GER, N3PS)
understood (TRAN, PAST and PASTP, GER, ANYS)
understanding (TRAN, PREP, GER, NOS)
understands (TRAN, PRE, GER, 3PS)
buy (TRAN, OR and PRE, NO2V, ANYS)
bought (TRAN, PAST and PASTP, NO2V, ANYS)
buying (TRAN, PREP, NO2V, NOS)
buys (TRAN, PRE, NO2V, 3PS)
go (INT, OR and PRE, INF/GER(D), N3PS)
went (INT, PAST, INF/GER(D), ANYS)
gone (INT, PASTP, INF/GER(D) , NOS)
going (INT, PREP, INF/GER(D), NOS)
goes (INT, PRE, INF/GER(D), 3PS)
come (INT, OR / PRE / PASTP, INF/GER(D), N3PS)

came (INT, PAST, INF/GER(D), ANYS)
coming (INT, PREP, INF/GER(D), NOS)
comes (INT, PRE, INF/GER(D), 3PS)

Personal pronouns

Lexical information (Number, Person, Case)

he (singular, 3rd person, subject case)
him (singular, 3rd person, object case)
I (singular, 1st person, subject case)
you (singular, 2nd person, subject and object case)
me (singular, 1st person, object case)
they (plural, 3rd person, subject case)
them (plural, 3rd person, object case)
this (singular, 3rd person, subject and object case)
Note: This word should officially be demonstrative pronoun. It is placed in this category for the convenience.

Nouns

Lexical information (Number, Person, Type of the first sound)

height (singular, neutral, non-vowel)
idea (singular, neutral, vowel)
ideas (plural, neutral, vowel)
book (singular, neutral, non-vowel)
books (plural, neutral, non-vowel)
head (singular, neutral, non-vowel)
word (singular, neutral, non-vowel)
words (plural, neutral, non-vowel)
student (singular, neutral, non-vowel)
students (plural, neutral, non-vowel)
walk (singular, neutral, non-vowel)
money (uncountable, neutral, non-vowel)
car (singular, neutral, non-vowel)
cars (plural, neutral, non-vowel)
boy (singular, male, non-vowel)
boys (plural, male, non-vowel)
girl (singular, female, non-vowel)
girls (plural, female, non-vowel)
father (singular, male, non-vowel)
fathers (plural, male, non-vowel)
mother (singular, female, non-vowel)
mothers (plural, female, non-vowel)
airplane (singular, neutral, vowel)
airplanes (plural, neutral, vowel)

Adjectives

Lexical information (Number of the noun that follows the word, Type of the first sound)

average (takes singular noun, vowel)
bright (takes singular/plural noun, non-vowel)
new (takes singular/plural noun, non-vowel)
happy (takes singular/plural noun, non-vowel)
great (takes singular/plural noun, non-vowel)
old (takes singular/plural noun, vowel)
extra (takes singular/plural noun, vowel)

Determiners

Lexical information (Number of the noun that should follow the word)

the (singular/plural noun)
a (singular noun)
an (singular noun)
several (plural noun)
many (plural noun)
this (singular noun)
these (singular noun)
one (singular noun)
three (plural noun)
every (singular noun)
his (singular/plural noun)
my (singular/plural noun)
your (singular/plural noun)
their (singular/plural noun)

Adverbs

about (no information)
not (no information)
very (no information)

Prepositions

in (no information)
with (no information)
to (no information)
about (no information)
for (no information)

Conjunction

and (no information)

References

Allen, J. 1987. *Natural Language Understanding*. Menlo Park, CA: Benjamin/Cummings.

Balcom, Patricia. 1997. Why is this happened? Passive morphology and unaccusativity. *Second Language Research,*13.1, pp. 1-9.

Cardie, Claire, Scott Mardis, and David Pierce. 1999. Combining error-driven pruning and classification for partial parsing. In *Proceeding of the Sixth International Conference on Machine Learning*, pp. 87-96.

Chandrasekaran, B. 1988. Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design. *IEEE Expert*, 1.3, pp. 23-30.

Clancey, W.J. 1985. Heuristic Classification. *Artificial Intelligence*, 27.3, pp. 289-350.

DeSmelt, William H. 1995. Ker Kommissar: an ICALL conversation simulator for intermediate German. In *Intelligent Language Tutors, Theory Shaping Technology*, ed. V. Mellisa Holland, Jonathan D Kaplan and Michelle R. Sams, pp. 153-174. New Jersey: Laurence Erlbaum Associates, Inc., Publishers.

Durkin, John. 1994. Expert Systems Design and Development. New York: Macmillan Publishing Company.

Ellis, Rod. 1986. Understanding Second Language Acquisition. Oxford; New Work: Oxford University Press.

Frank, Marcella. 1972. Modern English a practical reference guide. Englewood Cliffs, New Jersey: Regents/Prentice Hall.

Hornby, A.S. 1976. Guide to Patterns and Usage in English, second edition. Oxford; New York: Oxford University Press.

Jackson, Peter. 1986. *Introduction to Expert Systems*. Wokingham, England; Reading, Mass: Addison-Wesley Pub Co.

Loriz, Donald. 1995. GPARS: a suite of grammar assessment systems. In *Intelligent Language Tutors, Theory Shaping Technology*, ed. V. Mellisa Holland, Jonathan D Kaplan and Michelle R. Sams, pp. 121-133. New Jersey: Laurence Erlbaum Associates, Inc., Publishers.

Oshita, Hiroyuki. 2000. *What is happened* may not be what appears to be happening: a corpus study of 'passive' unaccusatives in L2 English. *Second Language Research,* 16.4, pp. 293-324.

Rich, Elaine and Kevin Knight. 1991. *Artificial Intelligent second edition*. New York: McGraw-Hill, Inc.

Roberstson, Daniel. 2000. Variability in the use of the English Article System by Chinese Learners of English. *Second Language Research,* 16.2, pp. 135-172.

Swan, Michael. 1980. *Practical English Usage*. Oxford; New York: Oxford University Press.

Wong, C.J. 1996. *Computer Grammar Checker and Teaching ESL Writing*. Paper presented at the 9th Annual Midlands Conference on Language and Literature, Omaha, NE. http://www.coe.missouri.edu/~cjw/portofolio/grammar-checker.htm