Robo-Sensei's NLP-Based Error Detection and Feedback Generation

Noriko Nagata
University of San Francisco

ABSTRACT

This paper presents a new version of Robo-Sensei's NLP (Natural Language Processing) system which updates the version currently available as the software package *ROBO-SENSEI: Personal Japanese Tutor* (Nagata, 2004). Robo-Sensei's NLP system includes a lexicon, a morphological generator, a word segmentor, a morphological parser, a syntactic parser, an error detector, and a feedback generator. The present paper focuses on the error detector and the feedback generator and describes how those components are integrated into Robo-Sensei's NLP system, what types of errors are detected, and what kinds of feedback messages are generated. The author is currently developing a new Robo-Sensei online textbook (Nagata, in press). The updated system is the NLP processing engine behind the new textbook.

KEYWORDS

Natural Language Processing, Error Detection, Feedback Generation, Japanese, Online Textbook

ROBO-SENSEI'S NLP SYSTEM

This paper presents the computational strategy behind a new, unreleased version of Robo-Sensei's natural language processing (NLP) system, which updates the version currently available as the software package *ROBO-SENSEI: Personal Japanese Tutor* (Nagata, 2004). Because the published version is intended as a supplement to be used along with regular printed textbooks, its handling of grammatical structures is not sufficiently comprehensive to cover the full Japanese curriculum. The new system can analyze all of the grammatical structures introduced in a standard 2- to 3-year Japanese curriculum. It will serve as the backbone of a new, online CALL Japanese textbook capable of providing immediate, personalized feedback in response to errors produced by students in full-sentence-production exercises. The present paper focuses on strategies for error detection and feedback generation and describes how these strategies are integrated into Robo-Sensei's NLP system, what types of errors are detected, and what kinds of feedback messages are generated.

In the past decade or so, there has been increasing interest in applying NLP to automatic error detection and feedback generation in CALL (e.g., Sanders, 1991; Swartz & Yazdani, 1992; Nagata, 1993, 2002; Holland, Kaplan, & Sams, 1995; Yang & Akahori, 1998; Dansuwan, Nishina, Akahori, & Shimizu, 2001; L'haire & Vandeventer Faltin, 2003; Reuer, 2003; Delmonte, 2003; Heift, 2003; Tschichold, 2003; Heift & Schulze, 2007), in addition to a number of NLP-based projects presented at the 2008 CALICO preconference workshop (e.g., Amaral & Meurers, 2008; De Felice & Pulman, 2008: Jang, Lee, & Seo, 2008; Oyama, Matsumoto, Asahara, & Sakata, 2008).

Robo-Sensei offers NLP-based Japanese language exercises in which students are provided with a communicative context and are asked to produce a sentence in response. The following is one of the exercise questions provided by *ROBO-SENSEI* Causative Lesson 2 Daily Chores (Nagata, 2004):

(1) Sample exercise

Context: Your teacher asked how your weekend was. You had a nice party for your father's birthday. Your mother cooked a lot, so you had to help her with cleaning dishes after the party. Tell the teacher that you were made by your mother to wash many big dishes.

The following sentence is a possible correct answer for this question.

(2) ははにおおきいおさらをたくさんあらわせられました。
Haha ni ookii osara o takusan arawaseraremashita.
'I was made by my mother to wash many big dishes.'

To see how many more correct answers are possible, the following characteristics of Japanese must be considered. Most words in Japanese can be written in either hiragana or kanji,[1] some words can be replaced by different lexical items to convey the same meaning, some words may be dropped when the context makes them clear,[2] and Japanese word order is relatively flexible.[3]

For possible correct answers, the above sentence can be multiplied by two alternative spellings of *haha* 'mother' (はは in hiragana and 母 in kanji), two alternative spellings of *ookii* 'big' (おおきい and 大きい), six alternatives for *osara* 'dishes' (おさら, お皿, 御さら, 御皿, さら, and 皿),[4] seven alternatives for *takusan* 'many' (たくさん, 沢山, いっぱい, 一杯, いくつも, なんまいも, and 何枚も),[5] two alternative spellings of *arawaseraremashita* (あらわせられました and 洗わせられました), three alternatives for *watashi wa* 'I (topic and subject)' (either dropping *watashi wa* as in the above sentence[6] or including *watashi wa* with the two spellings わたしは and 私は), and six alternative word orders (the word order among the three noun phrases, ははに *haha ni*, おおきいおさらを *ookii osara o*, and たくさん *takusan* could be shuffled). Lexical, orthographical, and word-order variants bring the total to 6,048 possible correct sentences:

(3) Possible correct sentences

ははにおおきいおさらをたくさんあらわせられました。 *Haha ni ookii osara o takusan arawaseraremashita.* x 2 alternatives for *haha* (はは, 母) x 2 alternatives for *ookii* (おおきい, 大きい) x 6 alternatives for *osara* (おさら, お皿, 御さら, 御皿, さら, 皿) x 7 alternatives for *takusan* (たくさん, 沢山, いっぱい, 一杯, いくつも, なんまいも, 何枚も) x 2 alternatives for *arawaseraremashita* (あらわせられました, 洗わせられました) x 3 alternatives for *watashi wa* (dropped one, わたしは, 私は) x 6 alternative word orders (e.g., ははにおおきいおさらをたくさんあらわせられました, ははにたくさんおおきいおさらをあらわせられました, おおきいおさらをははにたくさんあらわせられました, etc.) = 6,048 correct sentences

Next, consider the possible incorrect answers the student might produce. Just switching the particle *ni* attached to *haha* in the 6,048 correct sentences with three erroneous particles (e.g., *ga*, *o*, and *de*) or omitting the particle altogether, yields 24,192 incorrect sentences, a total that can be multiplied by at least five possible particles for the particle *o* attached to *osara* (i.e., the correct particle *o*, three erroneous particles, and one omission) and at least eight different conjugations of *arawaseraremashita* (i.e., the correct conjugation and seven mistaken conjugations changing the form to present, direct, and/or passive forms). The total number of possible incorrect sentences is 967,680.

It is now clear why it is hopeless to manually list all grammatical and ungrammatical sentences even for just one question (Nagata, in press). Therefore, it is also hopeless to manually associate an informative feedback message with each of the hundreds of thousands of possible input sentences. The point of Robo-Sensei's NLP technology is to feasibly overcome this combinatorial explosion in order to provide detailed and relevant feedback.

Analysis of a Correct Answer

Answer schema

For the question presented in (1), Robo-Sensei stores only the following answer schema instead of listing all of the numerous possible answers discussed above.

(4) Example of an answer schema

```
(わたし n+は p wa-topic+?私 n+は p wa-topic+?)
(はは n/母 n/)(に p ni-agCauser)
(おおきい a おおきい/大きい a 大きい/)
<(お f)(さら n)|(お f)(皿 n)|(御 f)(さら n)|(御 f)(皿 n)|(さら n)|(皿 n)|>
(を p o-obj)<(たくさん n)|(沢山n)|(いっぱい n)|(一杯 n)|(いくつ n)(も p mo-addition)|(なん n)(まい n)(も p mo-addition)|(何 n)(枚 n)(も p mo-addition)|>
(あらわ v あらう/洗わ v 洗う/)(せ x)(られ x)(ました x)
```

Robo-Sensei automatically generates relevant possible correct answers using both the answer schema and the student's response. It then parses the correct answers first and the student response second. There are no spaces between words in written Japanese texts. However, words in the correct answer schema are already segmented as seen in (4), so the word segmentor does not have to separate individual words. On the other hand, words in the student's answer are not separated (the student's answer is nothing but a string of characters), thus the word segmentor needs to segment the student string prior to parsing, according to the process described in the section on the analysis of a student answer below. The error detector and the feedback generator utilize detailed grammatical information derived from the parse: they compare the grammatical nature of the correct answer to that of the student response, locate differences between them, and generate relevant feedback messages based on those differences.

The following sequence illustrates this process. Suppose a student produces the sentence in (5) to respond to the question in (1):

(5) ははがおおきいのおさらたくさんをあらわれました 。
Haha ga ookii no osara takusan o arawaremashita.

Robo-Sensei first identifies lexical items the student did not use and eliminates them from the answer schema. For example, the student did not use the optional phrase わたしは, so it is eliminated from the answer schema. The student used the hiragana version of は, so the kanji version is eliminated, and so forth. The reduced answer schema looks like the schema in (6).

(6) Example of a reduced answer schema

(は n)(に p ni-agCauser)
(おおきい a おおきい)
(お f)(さら n)
(を p o-obj)(たくさん n)
(あらわ v あらう)(せ x)(られ x)(ました x)

Then, each lexical item in (6) is searched in the lexicon, and additional lexical information for each item stored in the lexicon is inserted in the answer schema.[7]

(7) Lexical information added to (6)

(は n ownFamily k-haha 11)(に p ni-agCauser)(おおきい a おおきい k-ookii aff imperf direct 3)(お f o-honor k-o 30)(さら n k-sara 30)(を p o-obj)(たくさん n degree)(あらわ v あらう u-verb ga-subj o-obj de-loc de-instr k-arau 1)(せ x せる ru-verb ga-causer ni-causee o-causee causative stem vowel 2 1)(られ x られる ru-verb ga-affected ni-agent niyotte-agent ga-kanooSubj ga-kanooObj ga-dekiruObj o-kanooObj poten-pass stem 2 2)(ました x ました masu-form aff perf distal 2)

Morphological parser

Next, this information is sent to the morphological parser, which consists of a set of morphological rules that build compounds and predicate conjugations. For example, if consecutive words follow compound rules, they are combined into a noun compound or a final predicate form. The result of the morphological parser in (8) shows that the polite prefix お *o* and the noun さら *sara* 'dishes' are combined as おさら *osara*. The verb base form あらわ *arawa* 'wash,' the causative form せ *se*, the passive form られ *rare*, and the formal form まして *mashita* are combined to construct the final verb あらわせられました *arawaseraremashita* 'was made to wash' in the causative passive formal past form.

(8) Result from the morphological parser

(は n ownFamily k-haha 11)(に p ni-agCauser)(おおきい a おおきい k-ookii aff imperf direct 3)(おさら n o-honor respect humble k-osara)(を p o-obj)(たくさん n degree)(あらわせられました v あらう u-verb o-obj de-loc de-instr k-arau ru-verb ga-causer ni-causee causative vowel ru-verb ga-affected ni-agent niyotte-agent ga-kanooSubj ga-kanooObj ga-dekiruObj o-kanooObj poten-pass masu-form aff perf distal)

Syntactic parser

Next, the morphological information is sent to the syntactic parser, which employs context free phrase structure rules to build words into phrases and phrases into sentences, using a bottom-up parsing technique (Winograd, 1983; Matsumoto, 1988). The author has developed fourteen PS (phrase structure) rules, and each PS rule is considered, in order, to check whether it applies to the input character string. For example, PS rule 1 NP → N (P) (P) is applied to identify the N (noun) はは *haha* and the P (particle) に *ni* as an NP (noun phrase), the N おさら *osara* and the P を *o* as an NP, and the N たくさん *takusan* as an NP.

(9) Result from Rule 1: NP → N (P) (P)

[NP はは(n ownFamily k-haha 11) に(p ni-agCauser)]
[A おおきい(aおおきい k-ookii aff imperf direct 3)]
[NP おさら(n o-honor respect humble k-osara) を(p o-obj)]
[NP たくさん(n degree)]
[V あらわせられました(v あらう u-verb o-obj de-loc de-instr k-arau ru-verb causative vowel ru-verb ga-affected ni-agent niyotte-agent passive aff perf distal)]

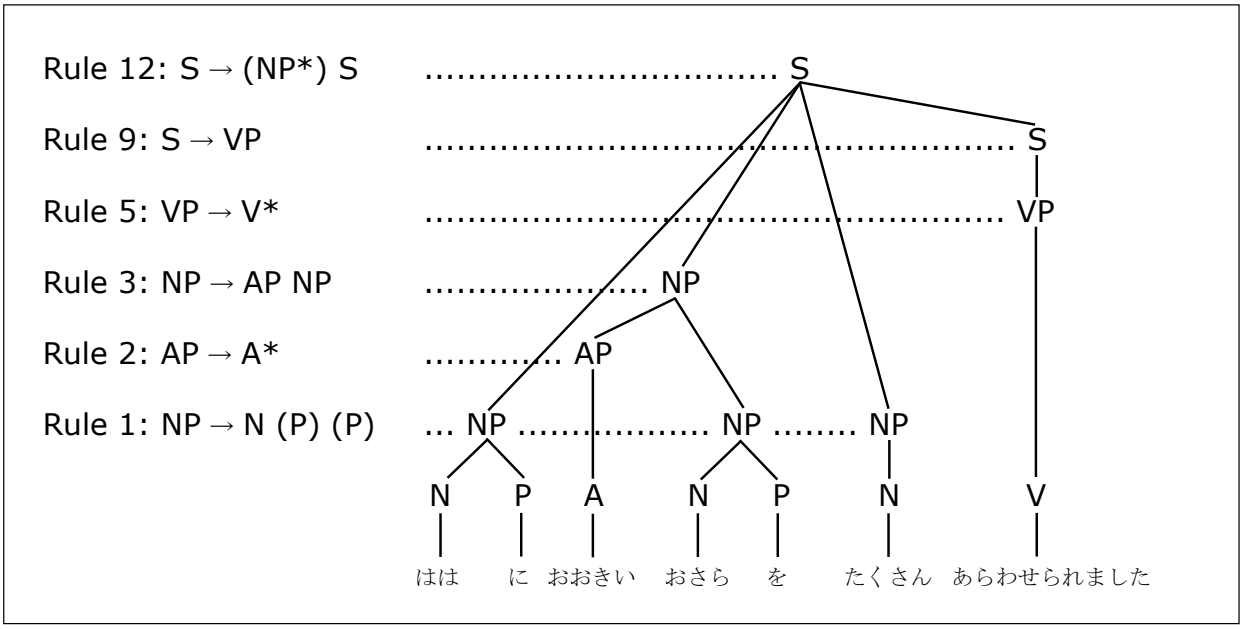
After all of the PS rules are checked, the grammatical information is obtained from the syntactic parser, verifying that the parser successfully built a grammatical sentence S, as seen in (10).

(10) Result from the syntactic parser

[S はは(n ownFamily k-haha 11) に(p ni-agCauser)
おおきい(a おおきい k-ookii aff imperf direct 3) おさら(n o-honor respect humble k-osara) を(p o-obj)
たくさん(n degree)
あらわせられました(v あらう u-verb o-obj de-loc de-instr k-arau ru-verb ga-causer ni-causee causative vowel ru-verb ga-affected ni-agent niyotte-agent ga-kanooSubj ga-kanooObj ga-dekiruObj o-kanooObj poten-pass masu-form aff perf distal)]

Figure 1 illustrates the parse tree of the correct answer. Note that only rules 1, 2, 3, 5, 9, and 12 were actually applied to build the parse tree.

Figure 1
Parse Tree of a Grammatical Sentence



Analysis of a Student Answer

Word segmentor

As described above, before analyzing a student answer, a correct answer is parsed and the detailed grammatical information is obtained from the syntactic parser. This grammatical information is now used to analyze the student's answer.

As also mentioned earlier, words are not segmented (separated by spaces) in written Japanese, so the first step in analyzing the student's answer is to segment the string into words. Word segmentation is a nontrivial problem due to the potential for multiple ambiguities. In the student answer in (5) ははがおおきいのおさらたくさんをあらわれました, the first character は *ha* and the second character は *ha* can be each interpreted as the particle *wa* or the two together as the noun はは *haha* 'mother.' The third character が *ga* can be the phrase particle *ga* or the clause particle *ga*. The string たくさん *takusan* can be segmented like たく *taku* as the ku-form of the desiderative *tai* and さん *san* as the number *san* 'three' or the suffix *san* 'Mr./Ms.,' or as the noun たくさん *takusan* 'many,' and so forth.

The word segmentor starts segmenting the character string from left to right and lists all possible segmentations. It is assumed as a heuristic that longer segments are more likely grammatical and intended than short, choppy segments: for example はは as a single noun is more likely as an intended interpretation than two successive particles *wa wa*, so the nominal interpretation is given priority.

In addition, to reduce ambiguity in segmentation, the following strategies are adopted. One is to prefer segmentations agreeing as much as possible with the answer schema. If the character the student used is not listed in the answer schema, any possible interpretation of the character has to be listed. For example, the student used an erroneous character が *ga*, and it is not listed in the answer schema, therefore both the phrase particle *ga* and the clause particle *ga* must be listed as possible interpretations. This illustrates how student errors com-

plicate the parsing process by introducing ambiguities not controlled by the answer schema. Another strategy for reducing alternative segmentations is to eliminate segmentations that already clearly violate grammatical structures. In fact, a Japanese sentence never starts with the phrase particle, so segmentations beginning with the phrase particle は *wa* are eliminated by this constraint. A third strategy for eliminating alternative segmentations is to drop one-syllable words from the lexicon (words written with one hiragana character) if they are not required for the exercise in question because one-hiragana-character words can result in a large number of alternative segmentations. For example, several verb roots consist of one hiragana each (e.g., で *de* the root form of でる *deru* 'attend' and み *mi* the root form of *miru* 'look at') as do several nouns (e.g., き *ki* 'tree' and せ se 'height'). These are suppressed from the lexicon but are put back into the lexicon whenever exercises require those words. The kanji versions of those one-hiragana words do not cause segmentation ambiguities like the hiragana versions, so the kanji versions stay in the lexicon all the time. The Japanese numerals include quite a few one-hiragana words (e.g., に ni 'two,' よ yo 'four,' ご go 'five,' etc.), so the numeric words are dropped from the lexicon unless the exercise requires number words. However, Japanese particles are never deleted from the lexicon, even though most of them are one-hiragana words (e.g., は *wa*, が *ga*, を *o*, に *ni*, で *de*, の *no*, etc.) because one of the main functions of the system is to analyze student particle errors. The preceding strategies can eliminate many incorrect segmentations of the student's input, but a number of alternative segmentations may remain.

Morphological parser

Alternative segmentations that are ranked as more plausible by virtue of the above heuristics are listed prior to less plausible segmentations. Then, they are sent in this order to the morphological parser for analysis. The morphological parser, consisting of rules that identify compounds and predicate conjugations, runs through all alternative segmentations in the list.

Prior to morphological processing, all prefixes, suffixes, and auxiliary forms are marked with an "x," but, if any prefixes, suffixes, and auxiliary forms are left uncombined after going through the morphological rules, they are labeled ungrammatical and the mark "x" is changed to "?." Then segmentations including any "?" notations are eliminated from further syntactic analysis. In this way, the morphological parser is capable of eliminating quite a few ungrammatical segmentations.

The two segmentations in (11) are derived from the morphological parser for the student answer under consideration. They differ only in the interpretation of が *ga* (the notation "p" is used for a phrase particle and "e" for a clause particle). Other examples may give rise to far more ambiguities.

(11) Result from the morphological parser (grammatical notations dropped for clarity)

Segmentation 1:
(はは n ...)(が p …)(おおきい a …)(の p …)(おさら n …)(たくさん n …)
(を p o-obj …)(あらわれました v …)

Segmentation 2:
(はは n ...)(が e …) (おおきい a …)(の p …)(おさら n …)(たくさん n …)
(を p o-obj …)(あらわれました v …)

Error detector and feedback generator

The result from the morphological parser is now sent to the syntactic parser. The error detector and the feedback generator are integrated into the syntactic parser. Error detection is more than just parsing since a parser only has to return "not well formed" when parsing fails. The whole point of error detection is to provide detailed, sympathetic feedback to students. Robo-Sensei's feedback not only indicates what is wrong but also explains why it is wrong: it explains the grammatical principles governing the errors to help students to apply those principles to produce the target sentence. This kind of feedback is crucial. For example, many students correct Japanese particle errors by a random, trial-and-error approach without understanding the principles behind proper usage. Unless students acquire the principles, they continue to use particles randomly and never master them. More generally, the pedagogical effectiveness of principle-based feedback has been demonstrated by a series of empirical studies (Nagata, 1993, 1995, 1996, 1997a, 1997b, 2007; Nagata & Swisher, 1995). The findings from the studies also agree with the cognitive approach of Omaggio (1986), "because language behavior is constantly innovative and varied, students must be taught to understand the rule system rather than be required to memorize surface strings in rote fashion" (p. 67).

Errors are classified into "unknown word," "missing word," "unexpected word," "predicate form error," "modifier error," and "word order error."

Unknown words. Before activating the syntactic parser, the error detector first checks whether there are any unknown words in the student answer which do not exist in the system lexicon. Recall that the word segmentor segments the string in the student answer. If no complete segmentation is created, every unsegmentable fragment is marked with "?." Also, even if the word segmentor can create a complete segmentation, if any prefixes, suffixes, or auxiliary forms are left uncombined by the morphological parser, they are marked with "?" as well. Then, if the error detector finds a "?" marked fragment in any segmentation, such a segmentation is neither sent to the syntactic parser nor subjected to any further error detection, and the message "unknown-word" is generated and stored in the message box. If every segmentation has a "?" marked fragment, the stored unknown-word error message is returned to the screen as final feedback.

For example, suppose that the student typed せら *sera* instead of さら *sara* 'dishes' in the question under consideration. The word segmentor identifies せ as a causative form and ら as a conditional form and creates a complete segmentation, but the morphological parser cannot combine せ andら into a correct verb form, so せ and ら are both marked with "?." There is no word like せら (and all nonsense words are treated as unknown words). The following error message appears on the screen: "The character string ..せら.. is unknown. Check your spelling."

Some common conjugation errors are handled not simply as unknown words in Robo-Sensei. For example, if the student typed あらせられました *araseraremashita* instead of あらわせられました *arawaseraremashita* 'was made to wash,' the error message is returned as follows: "Your causative conjugation あらせ is incorrect. Insert 'わ' after あら."

Robo-Sensei's new feedback generation system has been improved to handle a wider range of misspellings which used to be treated as "unknown words" in the previous version. Long vowels are often misspelled. For example, きょうと *Kyooto* is misspelled as きょと*Kyoto* in hiragana because in English it is spelled as *Kyoto*, not *Kyooto*. So, the error message appears as follows: "きょと is misspelled. Insert 'う' after きょ." Many Japanese adjectives end with a long vowel *ii*, and students often omit the last *i*. For instance, if students typed おおき *ooki* instead of おおきい *ookii* 'big,' the error message is pesented as "Your adjective form おおき is incorrect. The last い is missing." Therefore, typical long vowel misspelling errors are taken care of.

In Japanese, the first syllable of the double consonants must be spelled with the small っ *tsu*, but students often fail to recognize it as the small っ *tsu* in the written form and type the regular size つ *tsu*. In this case, the error message is "change つ to the small っ for the first syllable of the double consonants."

Missing words. Next, the error detector checks whether there are any missing words in the student answer. For example, if the student missed the word たくさん in a sentence, the error message is generated as "たくさん is missing."

Unexpected words. Next, the error detector checks whether there are any unexpected words in the student answer (i.e., words in the lexicon but not in the correct answer schema). When an unexpected word violates some special usage, additional error messages are generated. For example, if the student used the word おかあさん *okaasan* instead of はは *haha* (both mean 'mother'), the following error message is returned: "おかあさん is not expected to be used here. おかあさん is used to refer to someone else's family member when speaking with an out-group person." All family terms are notated with either "ownFamily" or "otherFamily" in the lexicon, and such lexical information is added to the answer schema (see [7] above) and to the student answer in the segmentation process. If the error detector finds the "otherFamily" notation in the student answer but not in the correct answer, then the feedback template in (12) is used to produce an error message.

(12) Example of a feedback template for family terms
- "<Word> is not expected to be used here. <Word> is used to refer to someone else's family member when speaking with an out-group person."

The placeholder <Word> in the template is filled with the incorrect family term the student used. Feedback templates are discussed in greater detail in the section on the feedback template below.

Predicate form errors. Next, the error detector checks whether there are any predicate form errors, including tense, negation, style, and all auxiliary forms such as causative, passive, desiderative, potential, conditional, extended predicate, consultative, honorifics, and so forth. For example, suppose the student typed あらわれました *arawaremashita* 'was washed' instead of あらわせられました *arawaseraremashita* 'was made (caused) to wash.' The result from the morphological parser for the correct answer is compared to that for the student's answer.

(13) Result from the morphological parser for the correct answer

…(あらわせられました v あらう u-verb o-obj de-loc de-instr k-arau ru-verb ga-causer ni-causee causative vowel ru-verb ga-affected ni-agent niyotte-agent …)

(14) Result from the morphological parser for the student answer

…(あらわれました v あらう u-verb o-obj de-loc de-instr k-arau ru-verb ga-affected ni-agent niyotte-agent passive …)

Here, the "causative" notation appears in correct answer (13), but not in student answer (14), so the error detector flags this discrepancy and the following error message is generated: "あらわれました is not in the causative form. Change it to the causative form."

Syntactic parser

At this stage, the student answer presented in (11) is sent to the syntactic parser. When there is more than one segmentation, as seen in (11), the first segmentation set is sent first for analysis. After the parsing of the first segmentation is complete, the second set is sent, and so forth.

In addition to the diagnostic procedures discussed in the preceding section, many others are integrated into the syntactic parser. The syntactic parser mostly checks whether or not the student answer is a syntactically well formed sentence, but semantic diagnostic procedures are also crucial. For example, both けんがすしを食べました 'Ken ate sushi' and すしがけんを食べました 'Sushi ate Ken' are syntactically well formed, but the latter sentence is semantically strange. To check whether the student answer fits with the context provided by the exercise question, additional error checking programs are activated after each PS rule.

In the example under consideration, the syntactic parser begins by assigning terminal symbols (e.g., N for a noun, A for an adjective, P for a particle, V for a verb, etc.) to the words in segmentation in (11).

(15) Student answer (Segmentation 1)

Assignment of the terminal symbols

[N ははn ownFamily k-haha 11)]
[P がp ga-subj …)]
[A おおきいa おおきい k-ookii aff imperf direct 3)]
[P のp no-modifier …)]
[N おさらn o-honor …)]
[N たくさんn degree)]
[P をp o-obj …)]
[V あらわれましたv あらう u-verb …)]

Note: The terminal symbols merely duplicate the lower case parts of speech within the segments.

Then, each PS rule is considered, in order, for applicability to build up a sentence from words to phrases and from phrases to a final sentence. For example, PS rule 1 is applied to combine an N (noun) and a P (particle) into an NP (noun phrase). As the result of applying PS rule 1, (17) presents an NP structure consisting of a single noun followed by a particle in square brackets. The succeeding PS rules are used to develop other types of phrase structures and eventually all phrases are combined into a single S if the sentence is grammatical, as seen in (10) and in Figure 1.

After phrases are combined with other phrases by PS rules, the system does not save the lower level phrasal structures any longer. Therefore, any diagnostic routines depending upon lower level parsing details must be applied immediately, which is why diagnostic routines are applied after every PS rule application.

(16) Correct answer

Result from PS rule 1: NP → N (P) (P)

[NP ははn ownFamily k-haha 11) に(p ni-agCauser)]
[A おおきい(a おおきい k-ookii aff imperf direct 3)]
[NP おさら(n o-honor …) を(p o-obj …)]
[NP たくさん(n degree)]
[V あらわせられました(v あらう u-verb …)]

(17) Student answer (Segmentation 1)

Result from PS rule 1: NP → N (P) (P)

[NP ははn ownFamily k-haha 11) が(p ga-subj …)]
[A おおきい(a おおきい k-ookii aff imperf …)]
[P の(p no-modifier …)]
[NP おさら(n o-honor …)]
[NP たくさん(n degree) を(p o-obj …)]
[V あらわれました(v あらう u-verb …)]

As seen in (16) and (17), はは *haha* is marked with the causative agent particle に *ni* in the correct answer, but はは *haha* is marked with the particle が *ga* in the student answer. おさら *sara* is marked with the object particle を *o* in the correct answer but is marked with no particle in the student answer. たくさん *takusan* is not marked with any particle in the correct answer but is marked with を *o* in the student answer. So, these differences flag errors and generate the following error messages:

(18) Particle error messages generated after PS rule 1

Student answer: ははがおおきいのおさらたくさんをあらわれました。
<Particle error>
- You used the particle が to mark はは. However, はは is the agent (causer) by whom someone is caused to do something in a causative passive sentence, and should be marked with the particle に.
- The particle を should be attached to おさら to indicate that おさら is the object that the action operates upon.
- たくさん is a noun describing manner or degree, and such nouns usually should not be followed by a particle. The particle を attached to たくさん should be deleted.

Next, PS rule 2 AP → A* is applied to combine any number of adjectives into an AP (adjective phrase) (the notation A* indicates that any number of items marked with A can be combined). PS rule 3 NP → AP NP is applied to combine an AP and an NP into an NP.

(19) Correct answer

Result from PS rule 3: NP → AP NP

[NP はは(n ownFamily k-haha 11) に(p ni-agCauser)]
[NP おおきい(a おおきいaff imperf …) おさら(n o-honor …) を(p o-obj)]
[NP たくさん(n degree)]
[V あらわせられました(v あらう u-verb …)]

(20) Student answer

Result from PS rule 3: NP → AP NP

[NP はは(n ownFamily k-haha 11) が(p ga-subj …)]
[AP おおきい(a おおきいaff imperf …)]
[P の(p no-modifier …)]
[NP おさら(n o-honor …)]
[NP たくさん(n degree) を(p o-obj o-loc o-causee …)]
[V あらわれました(v あらう u-verb …)]

As seen in (19), PS rule 3 is applied to combine the AP おおきい *ookii* and the NP おさらを *osara o* into the NP おおきいおさらを *ookii osara o* 'big dishes object-particle' in the correct answer. On the other hand, in the student answer in (20), PS rule 3 could not be applied to combine the AP おおきい *ookii* and the NP おさら *osara* due to the intervening particle の *no* between them. The syntactic parser catches this syntactic error and generates the following error message: "おおきい is an adjective and directly modifies a noun without any particle. Delete の attached to おおきい."

Next, PS rule 4 NP → NP* is applied to combine any number of noun phrases into a single NP (with the condition that an NP combined with the following NP must be a nominal modifier). The student answer does not include nominal modifiers, so PS rule 4 does not apply here.

To illustrate a nominal modifier example, suppose that the student produced a sentence like the one in (21) (in response to some other exercise). The syntactic parser flags syntactic errors related to nominal modifiers, if any. Since the particle の *no* is wrong and it should be な in (21), the error message returns as follows: "にぎやか is a na-adjective. Use the particle な to modify the following noun."

(21) デパートはにぎやかのとおりにありました 。
Depaato wa nigiyaka no toori ni arimashita.
'The department store was located on a lively street.'

Before moving to PS rule 5, a word order checking program is inserted to check the word order among noun phrases. For example, suppose the student produced the following sentence.

(22) すしやそばやてんぷらを食べました 。
Sushi ya soba ya tenpura o tabemashita.
'I ate sushi, soba, tempura, and so forth.'

The word order of すし *sushi*, そば *soba*, and てんぷら *tenpura* in (22) can be shuffled, but the last item C in pattern 「A や *ya* B や *ya* C」 'A, B, C, and so forth' has to be marked with

an appropriate particle to indicate the semantic role of the whole noun phrase (C should be marked withを in this example). So the word order checking program can accept different orders among A や B や C, but if C is marked with an erroneous particle, a relevant particle error message is produced. If C is marked with a correct particle but is placed in the wrong position like てんぷらをすしやそばや食べました *Tenpura o sushi ya soba ya tabemashita*, a corresponding error message is generated as follows: "てんぷら is in the wrong place. It should come after the や-marked noun. Follow the pattern「AやBやCを」."

Next, PS rule 5 VP → V* is applied to combine any number of verbs into a VP (verb phrase). For example, the verb あらわれました *arawaremashita* in the student answer becomes the VP.

PS rule 6 NP → VP NP handles relative clause constructions in which a verb clause modifies a noun phrase. The student answer does not include a relative clause, so PS rule 6 is not applied.

After PS rule 6, a relative clause word order checking program is activated to check the word order involved in the relative clause. In English, a head noun in the relative clause comes first, while in Japanese a head noun comes after the relative clause, so students often produce word order mistakes for the relative clause construction. To illustrate an example of relative clause errors, suppose the student produced a sentence like the one in (23).

(23) とんかつは食べたデパートでとてもおいしかったです。
Tonkatsu wa tabeta depaato de totemo oishikatta desu.
'The tonkatsu I ate in the department store was very delicious.'

The relative clause word order checking program detects errors and generates the following messages:

- In your sentence, 食べた modifies デパート. However, 食べた should modify とんかつ, so it should come immediately before とんかつ.
- デパート is part of the relative clause and should come before the head noun とんかつ."

Error detection and feedback generation continue until the last PS rule is considered for applicability.

It remains to explain how Robo-Sensei decides which generated feedback messages to return to the student. As described in the word segmentor section above, all possible segmentations are listed in order of plausibility and Robo-Sensei parses each segmentation in that order. After a given segmentation is parsed, the system counts the total number of error messages the segmentation gives rise to. Robo-Sensei always remembers the feedback corresponding to the segmentation whose parse gives rise to the fewest errors so far. If a segmentation is encountered that gives rise to no errors, Robo-Sensei returns "Correct!" immediately. Otherwise, after all segmentations have been considered, Robo-Sensei returns the feedback corresponding to the segmentation with the least number of errors overall.

Experience with the system has shown that if the first segmentation is not interpreted as a correct answer, the subsequent segmentations are very unlikely to be diagnosed as correct answers, which implies that listing segmentations in priority order reduces expected processing time devoted to ungrammatical segmentations. Also, the general tendency is that

segmentations lower in the list are likely to contain more grammatical errors. Therefore, the syntactic parser is programmed to process at most 25 segmentations.

When each of the first 25 segmentations involves errors, the segmentation that involves the least number of errors is interpreted as the student's intention. That approach is robust but not perfect because of the possibility of mistaken segmentations, as explained in the Word Segmentor section. Robo-Sensei's beta-testers attempt to type erroneous sentences in a wide variety of ways. Then, whenever error messages are encountered that are confusing or do not make sense, the author checks which segmentations generate those messages. If fragments of segmentations are found that are ungrammatical and would never be intended by the student, the following procedure is taken. The ungrammatical segmentation fragments are stored as "spurious" in the word segmentor program, and segmentations containing the fragments are eliminated from consideration in future runs of the program. For example, it was observed that the character string なかった *nakatta* is segmented not only as a word for 'did not exist' but also as the particle な *na* and the verb かった 'bought,' which is a mistake because the particle な cannot be directly followed by any verb. Accordingly, the ungrammatical fragment involving these two words is stored as "spurious" in the word segmentor program, and all segmentations including that fragment are eliminated from future analyses.[8] This strategy is ad hoc in the sense that a spurious segmentation fragment is caught only when unexpected error messages are presented, but it nevertheless avoids unnecessary and confusing error messages and speeds up the process of analysis.

The complex analysis for the student answer in (5) above is performed by Robo-Sensei in a fraction of a second. The feedback messages generated at each stage of parsing are assembled for final presentation to the student on the screen as follows:

(24) Feedback messages in response to the student answer in (5)

Student answer: ははがおおきいのおさらたくさんをあらわれました。 <Particle error> - You used the particle が to mark はは. However, はは is the agent (causer) by whom someone is caused to do something in a causative passive sentence, and should be marked with the particle に. - おおきい is an adjective and directly modifies a noun without any particle. Delete の attached to おおきい. - The particle を should be attached to おさら to indicate that おさら is the object that the action operates upon. - たくさん is a noun describing manner or degree, and such nouns usually should not be followed by a particle. The particle を attached to たくさん should be deleted. <Predicate error> - あらわれました is not in the causative form. Change it to the causative form. Try it again!

Feedback template

One more feature to mention here is that Robo-Sensei's NLP system includes a number of feedback templates to produce error messages in a general way. That is to say, when errors are detected, the grammatical information derived from parsing is sent to the relevant feedback template and the template is filled with the grammatical information for final presentation. Templates are classified by types of errors. For example, the following feedback template is invoked for mistaken use of a particle.

(25) Example of a feedback template for particle errors
- You used the particle <P1> to mark <N> as if it were <P1-function>. However, <N> is <P2-function> and should be marked with the particle <P2>.

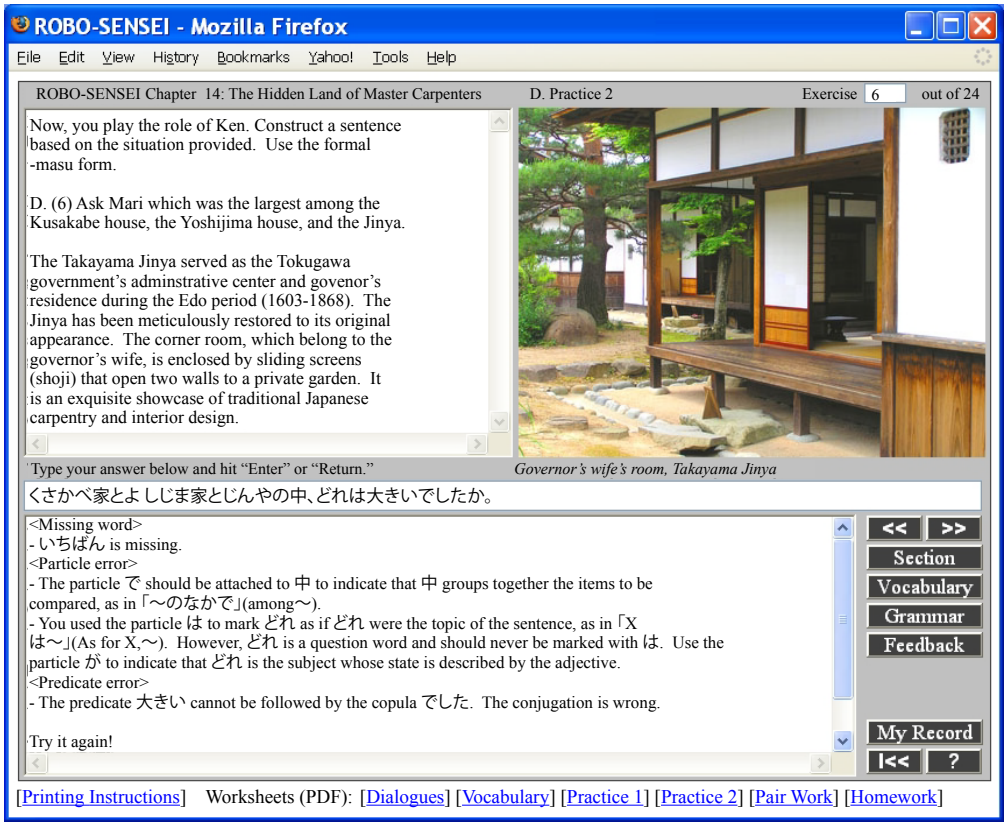
In the template in (25), <P1> is filled with the mistaken particle the student used, <P2> with the correct particle, <P1-function> and <P2-function> with the particle descriptions of P1 and P2, respectively, and <N> with the target noun attached to the particle in question. Suppose the student was asked to write a Japanese sentence equivalent to "I bought it in a department store" and produced a sentence like デパートをかいました *Depaato o kaimashita* 'I bought a department store.' The above template generates the corresponding error message as follows:

(26) Example of a feedback message presented to the student
- You used the particle を to mark デパート as if it were the object that the action operates upon. However, デパート is the location where the activity takes place, and should be marked with the particle で.

Screen image

The author is currently developing a new Robo-Sensei online textbook (Nagata, in press). The updated system described in the present paper is the NLP engine behind the new textbook. The following screen image provides an idea of what an exercise in the new Robo-Sensei online textbook looks like and how NLP-based grammatical feedback is presented in response to student errors. The exercise is taken from Chapter 14 The Hidden Land of Master Carpenters (Comparisons).

Figure 2
Screen Image of an Exercise in Chapter 14 The Hidden Land of Master Carpenters (Comparisons)



CONCLUSION

In this paper, it was explained how hopeless it is to list all correct and incorrect sentences even for just one question (there are hundreds of thousands of possible answers) and how infeasible it is to associate "canned" feedback messages with each possible answer. It was then explained how Robo-Sensei overcomes these problems by using NLP technology. Robo-Sensei stores a single, simple answer schema for each exercise instead of manually listing numerous possible answers. It automatically generates a set of relevant possible correct answers based on the answer schema and the student's response. Next, it parses both the correct answers and the student response, obtains detailed grammatical information from the parse, compares the grammatical nature of the correct answers to that of the student response, flags any differences between them, and generates principle-based feedback messages in response to student errors based on those differences. Robo-Sensei's key advantages in second language learning are its NLP parsing engine and its capacity to detect errors and generate feedback.

NOTES

[1] A Hiragana character is purely phonetic and is always read as one syllable. A Kanji character symbolizes a word and may be read with more than one syllable. For example, the word for 'mountain' in Japanese is *yama,* which can be simply written with two Hiragana characters や and ま or with one kanji character 山.

[2] For example, a sentence like 'I bought it' can be uttered as かいました *kaimashita* 'bought' in Japanese, so a single verb can be a whole sentence in Japanese.

[3] In Japanese, the predicate (verb, adjective, or copula) comes at the end of the sentence. Other than that, the order of noun phrases can be flexible because particles attached to noun phrases indicate their grammatical roles (e.g., subject, object, goal, etc.). The positions at which they appear in a sentence are not grammatical indicators, unlike English in which the subject comes first, the verb second, and the object third.

[4] *Osara* おさら consists of *o* お (polite prefix) and *sara* さら (dishes). The polite prefix is optional in this case. The kanji versions of お and さら are 御 and 皿, respectively.

[5] *Takusan* (たくさん, 沢山), *ippai* (いっぱい, 一杯), *ikutsu mo* (いくつも), and *nanmai mo* (なんまいも, 何枚も) mean 'many'. They are all acceptable.

[6] The expression w*atashi wa* is the topic and subject in this sentence and is optional for this question.

[7] Lexical items are associated with grammatical notations in the lexicon (e.g., aff, imperf, u-verb, ga-subj, etc.). These notations are used in the parsing process.

[8] At present, there are around 30 "spurious" segmentations stored in Robo-Sensei's word segmentation program.

REFERENCES

Amaral, L., & Meurers, D. (2008, March). *Little things With big effects: On the identification and interpretation of tokens for error diagnosis in ICALL.* Paper presented at the CALICO Preconference Workshop on Automatic Analysis of Learner Language, San Francisco, CA.

Dansuwan, S., Nishina, K., Akahori, K., & Shimizu, Y. (2001). Development and evaluation of a Thai learning system on the web using natural language processing. *CALICO Journal, 19*, 67-88. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

Delmonte, R. (2003). Linguistic knowledge and reasoning for error diagnosis and feedback generation. *CALICO Journal, 20*, 513-532. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

De Felice, R., & Pulman, S. G. (2008, March). *Automatic detection of preposition errors in learner writing.* Paper presented at the CALICO Preconference Workshop on Automatic Analysis of Learner Language, San Francisco, CA.

Heift, T. (2003). Multiple learner errors and meaningful feedback: A challenge for ICALL systems. *CALICO Journal, 20*, 533-548. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

Heift, T., & Schulze, M. (2007). *Errors and intelligence in computer-assisted language learning: Parsers and pedagogues*. New York: Routledge.

Holland, V. M., Kaplan, J. D., & Sams, M. R. (1995). *Intelligent language tutors*. Mahwah, NJ: Lawrence Erlbaum.

Jang, S. B., Lee, S., & Seo, S. (2008, March). *Annotation of Korean learner corpora for particle error detection.* Paper presented at the CALICO Preconference Workshop on Automatic Analysis of Learner Language, San Francisco, CA.

L'haire, S., & Vandeventer Faltin, A. (2003). Error diagnosis in the FreeText project. *CALICO Journal, 20*, 481-495. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

Matsumoto, Y. (1988). 自然言語理解 [Natural Language Understanding]. In H. Tanaka & J/ Tsujii (Eds.), 三章統語解析の手法 [Chapter 3: Parsing methods]. Tokyo: オーム社.

Nagata, N. (1993). Intelligent computer feedback for second language instruction. *The Modern Language Journal,* 77, 330-339.

Nagata, N. (1995). An effective application of natural language processing in second language instruction. *CALICO Journal, 13*, 47-67. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

Nagata, N. (1996). Computer vs. workbook instruction in second language acquisition. *CALICO Journal*, *14*, 53-75. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

Nagata, N. (1997a). The effectiveness of computer-assisted metalinguistic instruction: A case study in Japanese. *Foreign Language Annals*, *30*, 187-200.

Nagata, N. (1997b). An experimental comparison of deductive and inductive feedback generated by a simple parser. *System,* 25, 515-534.

Nagata, N. (2002). BANZAI: An application of natural language processing to web-based language learning. *CALICO Journal, 19*, 583-999. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

Nagata, N. (2004). *ROBO-SENSEI: Personal Japanese tutor*. Boston, MA: Cheng and Tsui.

Nagata, N. (2007). Input enhancement by natural language processing. In C. Gascoigne (Ed.), *Assessing the impact of input enhancement in second language education: Evaluation in theory, research and practice* (pp. 153-167). Stillwater, OK: New Forums Press.

Nagata, N. (in press). An online Japanese textbook with natural language processing. In I. Lancashire (Ed.), *Teaching literature and language online*. New York: MLA.

Nagata, N., & Swisher, V. (1995). A study of consciousness-raising by computer: The effect of metalinguistic feedback on second language learning. *Foreign Language Annals*, *28*, 337-347.

Omaggio, A. C. (1986). *Teaching language in context*. Boston: Heinle and Heinle.

Oyama, H., Matsumoto, Y., Asahara, M., & Sakata, K. (2008, March). *Construction of an error information tagged corpus of Japanese language learners and automatic error detection.* Paper presented at the CALICO Preconference Workshop on Automatic Analysis of Learner Language, San Francisco, CA.

Reuer, V. (2003). Error recognition and feedback with lexical functional grammar. *CALICO Journal, 20*, 497-512. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

Sanders, R. (1991). Error analysis in purely syntactic parsing of free input: The example of German. *CALICO Journal, 9*, 72-89. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

Swartz, M. L., & Yazdani, M. (Eds.). (1992). *Intelligent tutoring systems for foreign language learning: The bridge to international communication*. London: Longman.

Tschichold, C. (2003). Lexically driven error detection and correction. *CALICO Journal, 20*, 549-559. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

Winograd, T. (1983). *Language as a cognitive process. Vol. I: Syntax.* Boston: Addison-Wesley.

Yang, J. C., & Akahori, K. (1998). Error analysis in Japanese writing and its implementation in a computer assisted language learning system on the world wide web. *CALICO Journal, 15*, 47-66. Retrieved March 23, 2009, from https://calico.org/page.php?id=5

ACKNOWLEDGMENTS

This paper is based on the author's presentation at the 2008 CALICO preconference workshop on "Automatic Analysis of Learner Language: Bridging Foreign Language Teaching Needs." The author wishes to thank Detmar Meurers and Anne Rimrott for organizing a very successful workshop.

AUTHOR'S BIODATA

Noriko Nagata is Professor and Chair of the Department of Modern and Classical Languages and Director of Japanese Studies at the University of San Francisco. Her main research agenda is to apply natural language processing to CALL. She is currently working on expanding the NLP-based software package, *ROBO-SENSEI: Personal Japanese Tutor* to a stand-alone online Japanese textbook. Her publications also include a series of empirical studies examining the effectiveness of various CALL features in second language acquisition.

AUTHOR'S ADDRESS

Noriko Nagata
Department of Modern and Classical Languages
University of San Francisco
2130 Fulton Street
San Francisco, CA 94117
Phone: 415 422 6227
Fax: 415 422 6928
Email: nagatan@usfca.edu