ReCALL **13** (1): 110–120. Printed in the United Kingdom
© 2001 Cambridge University Press

Creating a grammar checker for CALL by constraint relaxation: a feasibility study

ANNE VANDEVENTER
LATL, Département de linguistique, Université de Genève, 1211 Genève 4, Switzerland
(*e-mail: Anne.Vandeventer@lettres.unige.ch*)

Abstract

Intelligent feedback on learners' full written sentence productions requires the use of Natural Language Processing (NLP) tools and, in particular, of a diagnosis system. Most syntactic parsers, on which grammar checkers are based, are designed to parse grammatical sentences and/or native speaker productions. They are therefore not necessarily suitable for language learners. In this paper, we concentrate on the transformation of a French syntactic parser into a grammar checker geared towards intermediate to advanced learners of French. Several techniques are envisaged to allow the parser to handle ill-formed input, including constraint relaxation. By the very nature of this technique, parsers can generate complete analyses for ungrammatical sentences. Proper labelling of where the analysis has been able to proceed thanks to a specific constraint relaxation forms the basis of the error diagnosis. Parsers with relaxed constraints tend to produce more complete, although incorrect, analyses for grammatical sentences, and several complete analyses for ungrammatical sentences. This increased number of analyses per sentence has one major drawback: it slows down the system and requires more memory. An experiment was conducted to observe the behaviour of our parser in the context of constraint relaxation. Three specific constraints, agreement in number, gender, and person, were selected and relaxed in different combinations. A learner corpus was parsed with each combination. The evolution of the number of correct diagnoses and of parsing speed, among other factors, were monitored. We then evaluated, by comparing the results, whether large scale constraint relaxation is a viable option to transform our syntactic parser into an efficient grammar checker for CALL.

1 Introduction

The main objective of the FreeText[1] project is to develop a Computer Assisted Language Learning (CALL) system for French as a foreign language, geared towards intermediate to advanced learners, following communicative approaches to Second Language Acquisition (SLA), and using Natural Language Processing (NLP) tools. Providing intelligent feedback on learners' full written productions in CALL software is a difficult task requiring an error diagnosis system. Such a system should ideally include a spell

[1]The FreeText project receives financial support from the European Commission in the IST programme of the 5th framework-programme, contract IST-1999-13093. The Swiss participation is funded by the Swiss Federal Office for Education and Science, contract 99.0049. The content of

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

checker, a grammar checker and a 'semantic' checker. Unfortunately, most checkers are designed for native speakers. "Teachers and linguists know only too well how different the errors made by native and non-native speakers are" (Granger & Meunier, 1994: 80). We have therefore decided to build our own diagnosis system.

The error diagnosis system we are implementing is composed of a spell checker, a grammar checker and a 'semantic' checker. It is designed to be integrated into CALL software and is accompanied by other tools and reference materials, such as a reference grammar and the sentence structure viewer *FipsGram* (see section 2.2 and the references mentioned there). The 'semantic' checker mostly verifies that the answer corresponds to the question and that the learner is not trying to fool the system with a grammatical sentence which has no relation with the current exercise.

The grammar checker component of this diagnosis system is based on a syntactic parser for grammatical French sentences. Because creating a robust syntactic parser with a large coverage is very time consuming, we are adapting a pre-existing one. Several techniques are envisaged to allow the parser to handle ill-formed input and to provide a diagnosis on this input. This paper describes an experiment which has been conducted to see whether a specific technique, constraint relaxation, could be effectively used in the present case.

The structure of this paper is as follows: Section 2 describes the syntactic parser and some of its current uses. The following section presents different techniques for diagnosing grammar mistakes. In section 4, the experiment is detailed. Results are given in section 5. Section 6 serves as a conclusion.

2 Syntactic parser

The *Fips* syntactic parser was chosen for the FreeText project because it was easily available to the research team. *Fips* has been fully described elsewhere (Laenzlinger & Wehrli, 1991; Wehrli, 1997), but we nevertheless present the characteristics which are important for a diagnosis system and for the transformation from a parser into a diagnosis system. The *Fips* parser is also an integral part of other applications which are briefly presented below.

2.1 The Fips parser

Fips is a syntactic parser based on the Government and Binding Theory (Chomsky, 1981; Haegeman, 1994, among others). The main goal of the parser is to provide syntactic structures for input sentences. The parsing algorithm is a chart parser and alternatives are treated in a pseudo-parallel way. Scores are given to each structure in order to classify alternatives during parsing and to choose which full analysis to display when parsing is successful. When no complete analysis is found by the parser, it presents partial analyses of the input, also called chunks. Thus, an output is provided even when the

this paper is the sole responsibility of its author and does not represent the opinion of the Community. The Community is not responsible for any use that might be made of the data appearing in this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

input does not result in a successful parse. The presence of partial analyses implies that the input is either non-grammatical or lies outside the capabilities of the parser. The parsing process is done in three main steps: (i) projection of structures following the X-bar model on the basis of lexical elements; (ii) combination of these structures; and (iii) application of constraints to reject non-grammatical structures. For the input in (1a), the structure in (1b) is proposed by *Fips*:

(1a) *Le sexisme est un phénomène très répandu.*
The sexism is a phenomenon very widespread

(1b) [TP[DP *le* [NP *sexisme*]][T' *est* [VP [DP *un* [NP *phénomène* [AP[DP e_i][AdvP *très*] [A' *répandu*]]]i]]]]

The labels used have the following meaning: TP is the tensed phrase, DP is the determiner phrase in the sense of Abney (1987), NP is the noun phrase, VP the verb phrase, AP the adjective phrase and AdvP the adverb phrase. 'e' corresponds to an empty category or to the trace of a moved element, following the Government and Binding Theory. Indexes are used to link two syntactically or semantically close elements, such as a pronoun and its referent.

2.2 Current usage

Besides giving syntactic structures for input sentences, *Fips* is also an integral part of several other applications. *FipsTag* is a tagger which provides each word of the input with a tag containing lexical and morpho-syntactic information. A large part of the information comes naturally from the lexicon, but lexical ambiguities are resolved through *Fips*' syntactic analysis. The *Fips* parser is also employed as part of translation tools to retrieve the syntactic information needed for translation from French into other languages (Etchegoyhen & Wehrli, 1998). Syntactic information provided by *Fips* is also used in a text-to-speech system, *FipsVox*, to reduce ambiguities, calculate liaison and elision phenomena, for example (Gaudinat, Goldman & Wehrli, 1998). A graphical output of *Fips* has also been created under the name *FipsGram* (Hamel & Vandeventer, 2000; Vandeventer, 1999). *FipsGram* is designed to give syntactic and lexical information to laymen and language learners. It is interactive in the sense that users can decide the kind of information they want to receive. The input sentence appears in a window where main grammatical functions such as subject, predicate and direct object can be underlined and individual words can be highlighted in a colour specific to their lexical category.

3 Diagnosis techniques

There are several diagnosis techniques available. Some are compatible with our parser and some are not. Some might combine one with another, others are independent. We briefly describe some techniques below and go into more details for constraint relaxation, the technique we studied in the experiment described in this paper.

Pattern matching was one of the first techniques to be used. It proceeds by comparison

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

of the input with one or several pre-stored answers. Different wrong pre-stored answers are needed to provide feedback for different types of errors. Although it can be quite complex with the use of logical expressions to express alternatives, pattern matching always requires comparisons to recorded answers. There is therefore no possible generalisation and, being hard coded, this technique involves a lot of work for each and every question one wants to be able to correct.

Error grammars are another possibility for diagnosis. This technique is used, among others, in the RECALL system (Murphy, Krüger & Grieszl, 1998) and the ICICLE system (Schneider & McCoy, 1998). It requires a parsing algorithm separate from the grammar and two sets of grammar rules, one containing the grammatical rules and the other the mal-rules to treat the errors. Problems raised by this approach are summarised by Matthews: "Of course, it has the drawback of increasing the size of the rule set with the attendant efficiency problems noted in the first section. It is also a problematic exercise in writing enough rules to handle all possible ill-formed input. Finally, there is also the problem in that simply producing a rule to allow through an ill-formed example does not provide an explanation for that failure" (Matthews, 1993: 19).

Because learners' texts tend to contain grammar errors which would make a parser fail, some researchers have taken the option of not parsing sentences completely, but of looking for errors with finite state automata. This was done, for example, by the *Laboratoire de Traitement du Langage et de la Parole* from the University of Neuchâtel who realised the ARCTA project (see among other Tschumi, Bodmer, Cornu, Grosjean, Kübler & Tschichold, 1994; and Kübler & Cornu, 1994). Each automaton is specialised and detects one type of error. Only errors for which an automaton has been created can therefore be corrected.

Some parsers, such as *Fips*, return partial analyses of sentences on which the parsing process fails to provide a complete analysis. Ideally, such a parser is also a recogniser: it accepts or rejects sentences, indicating whether they are grammatical or not. Therefore, any sentence which receives partial analyses is considered to contain at least one error. In examining the chunks provided, one can hope to deduce what type of error occurred and prevented a complete analysis.

Constraint relaxation is one of the most widely used diagnosis techniques and there are several versions of it. To name only a few, Weischedel & Black (1980:108) spoke of 'failable' constraints, Matthews (1993) of principle violation, and Heinecke, Kunze, Menzel & Schröder (1998) of graded constraints. Although there are some differences between these versions of the technique, the underlying principles are the same. Partial structures can combine only if some constraints or conditions are met, such as subject-verb agreement for number and person. When the constraints are relaxed, attachment is allowed even if the constraint is not respected. To be useful as a diagnosis system, the relaxed constraint must be marked on the structure, in order to be able to indicate later the type and position of the error detected. Proper labelling of where the analysis has been able to proceed thanks to the specific relaxation of a constraint forms the basis of the error diagnosis (Felshin, 1995). Most systems using constraint relaxation can relax constraints separately from each other and relax them only when they cannot be satisfied. One of the advantages of constraint relaxation is that it provides both a diagnosis and a complete analysis of the input sentence. The analysis can then be reused for other applications, such as 'semantic' checking. The disadvantage of this technique is that

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

parsers with relaxed constraints tend to produce more complete, although incorrect, analyses for grammatical sentences, and several complete analyses for ungrammatical sentences. This increased number of analyses per sentence has one major drawback: it slows down the system and requires more memory.

4 Experiment

The main objective of the experiment we conducted was to determine whether the constraint relaxation technique could be used to transform the *Fips* parser into a diagnosis system. Theoretically, the parsing algorithm allows it, as it uses constraints to reject non-grammatical structures. Nevertheless, it was necessary to check if, when applied to a large coverage system, the results were satisfying as to the quality of the diagnosis and as to the complexity of the algorithm. We selected three constraints which were relaxed and we observed *Fips*' behaviour with and without constraint relaxation.

4.1 Corpus

The corpus which was used for this experiment is an extract of the 'French Interlanguage Database'[2] (FRIDA), containing texts from learners of French. The chosen extract contains 21 texts written in exam conditions by Bulgarian learners of French. The subject of the essays is equality between men and women. The *Fips* parser counted 9010 words and 416 sentences in this extract, giving an average sentence length of 21.7 words. Example (2) presents two consecutive sentences, separated by periods, taken from the corpus.

(2) Aujourd'hui être homme ou femme à certains avantages inconvénients, mais être p. ex. un homme ne peux pas être avantage. Aujourd'hui une égalité entre les homme et les femmes il y a des femmes qui occupent des postes très importants (présidents ministres etc.) et elles ont prises la place qu'elle mérite dans la société.

Today be man or woman to some advantages disadvantages, but be e.g. a man NEG can not be advantage. Today an equality between the man and the women there are some women who occupy some positions very important (presidents ministers etc.) and they have taken the place that she merits in the society.

These two sentences contain some agreement mistakes, one sentence is incomplete, some words are missing, punctuation is far from perfect. The gloss provided is literal and word-to-word, errors are therefore not well represented. It is also difficult to say whether this sentence is representative of the whole corpus, but it shows some of the errors we will be confronted with.

[2]Use of the FRIDA corpus was graciously granted by the Centre for English Corpus Linguistics (CECL), Université Catholique de Louvain, Belgium. In the FreeText project, for which we are partners, the CECL is collecting a corpus of learner French, analysing it and designing an error typology. Their results were unfortunately not yet available when this experiment was conducted.

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

4.2 Constraints

For this experiment, we have decided to relax three constraints which are considered particularly suitable for relaxation in literature (Matthews, 1993; Tschichold, 1999), that is agreement in gender, agreement in number, and agreement in person. These constraints have the advantage of belonging to a well-defined domain, in which learners make mistakes.[3] Moreover, this type of mistake is easily identifiable and we could serve as experts on the corpus to manually identify all the errors linked to these three constraints. Each constraint can be relaxed individually or in combination with the others. By choosing three constraints that depend somewhat on one another, we hoped to observe interactions between the three constraints when they are all relaxed together. This should help us to better judge the system's decrease in efficiency when the three constraints are relaxed at the same time. However, we take the risk that the results of the experiment are not as representative as possible of all the relaxation combinations.

4.3 Technical details

To relax agreement constraints, we have modified agreement checking procedures in *Fips*. Agreement features are coded with sets, one for gender, one for number, and one for person. We use sets as it is possible for a single structure or word to have two values of the same set at the same time. The determiner *les* (the plural) is both masculine and feminine, for example. Without constraint relaxation, agreement, and therefore the combination of two structures, is only possible if there is a non-empty intersection for each pair of the three pairs of feature sets. In examples (3) and (4) below, only the gender agreement features are given, so as not to overcharge the representation; sets are represented within curly brackets '{}'.

(3a) [DP *l'*{masculine, feminine}]
the (masculine and feminine form)
(3b) [NP *égalité* {feminine}]
equality
(3c) [DP *l'*[NP *égalité*]{feminine}]
the equality

An agreement constraint relaxation is activated only if the intersection of the two sets in (a) and (b) is empty, that is only if the constraint cannot be satisfied. In the case illustrated in (4), the constraint is relaxed and the two structures can then combine. The agreement feature set of the resulting structure is not the intersection of the sets of the two structures, as it would be empty, but the union of those sets. Indeed, it is not always easy to determine what the learner is trying to express. In (4) below, the learner can be speaking about a woman doctor, in which case the feminine word *doctoresse* could be used, or s/he can be mistaken about the gender of the word *docteur*.

[3]First results from the CECL, just available at the time of writing, indicate that agreement in gender, number and person errors represent roughly 15% of the total errors in the corpus.

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

(4a) [DP *la* {feminine}]
the (feminine only form)
(4b) [NP *docteur* {masculine}]
doctor
(4c) [DP *la* [NP *docteur*]{masculine, feminine}<gender error>]
the doctor

Relaxing constraints to allow structures to combine is not useful for error diagnosis if one does not mark in some way which constraint has been relaxed and where. In the prototype used in the experiment, a note was added to the structure, as schematised in (4c), within the angle brackets '< >'. The notes containing the error diagnosis are passed along to another part of the CALL software. The form of the feedback and its nature are controlled outside of the diagnosis tool, as parameters of specific exercises and of the individual learner influence the kind of feedback that is given. Feedback can take the form of a correctly spelled word, an explanatory note, or a hyperlink to a tutorial or to the reference grammar.

5 Results

The corpus was fed to the error diagnosis prototype eight times. This gave us an analysis of the corpus for each combination of the three relaxed constraints, gender, number and person, as well as without any relaxation. The output of these analyses have been hand-tagged and compared to the corpus. We noted errors which were correctly diagnosed by the system, errors which were not diagnosed by the system, and finally errors which were detected by the system where there was none in the corpus, that is false positives. We first look at the results when the three constraints are relaxed at the same time. Then we examine the false positives in more details. Finally we summarise the other results.

5.1 Three constraints relaxed

A quantitative analysis of the output of the parser with the three mentioned constraints relaxed together gave the results summarised in Table 1. If recall is good, even perfect

Table 1. *Errors detected when relaxing the three constraints together*

Errors	Gender	Number	Person	Total
In the corpus	49	33	14	96
Detected correctly	41	23	14	78
Not detected	8	10	0	18
False positives	12	30	30	72
Precision in %[a]	77,4	43,4	31,8	52,0
Recall in %[b]	83,7	69,7	100	81,3

[a]Precision = errors detected correctly / (errors detected correctly + false positives) × 100
[b]Recall = errors detected correctly / errors in the corpus × 100

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

Table 2. *Details of the repartition of false positives*

False positives	Gender	Number	Person	Total
Lexical ambiguities	1	7	13	21
Misspelled words	1	2	1	4
Incorrect analyses	5	15	11	31
Incomplete analyses	2	4	2	8
Others	3	2	3	8
Total	12	30	30	72

for person agreement, precision is quite low. For error diagnosis in general, and even more so in CALL, precision is of the outmost importance. "Overflagging is what users of grammar checkers object to before all other things. It is a waste of users' time and leaves a bad impression of the system's capabilities. In addition, learners of a language cannot be expected to have reliable intuitions to fall back on when they see a wrong flag on a correct structure" (Tschichold, 1999: 213). Overflagging increases the risk of learning wrong structures on the part of language learners.

5.2 False positives

As it is very important to diminish the number of false positives as much as possible, we have taken a closer look at them in order to try to detect regularities to help us eliminate some of the overflagging. The results of this second examination are given in Table 2.

In the 'lexical ambiguity' category are found sentences for which the parser followed a wrong option for a word or a group of words lexically ambiguous. It is often a plural common noun which is taken for a verb in the present tense, at the second person of the singular, such as *avantages* (advantages). This type of false positive could perhaps be avoided if syntactic context was taken into account before relaxing the constraints.

Some words are misspelled. They were not detected by the spell checker because their misspelled form existed in the lexicon, with another meaning and often with another lexical category, the latter being very damaging for the analysis. This type of difficulty will be difficult to eliminate. One can try to use phonetics, but one must not use the result of the syntactic analysis to determine how the words are pronounced, as it is done in *FipsVox*.

In the 'incorrect analyses' category are grouped sentences for which the parser has not provided the preferred structure. Sometimes the problem lies with the attachment of an adjectival group, sometimes a wrong interpretation of which nominal group acts as the subject of the sentence, or a wrong scope for co-ordination. Work for this type of false positive is linked more to the parser itself than to the relaxation of constraints. Given that this category represents 43% of the false positives, it is essential work on it.

Notwithstanding the three relaxed constraints, the parser is not able to give a full analysis of all the sentences in the corpus, sometimes because of other errors in those sentences. When a sentence is not fully analysed, it is possible that the phrases required

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

for agreement are found in two separate chunks. This sometimes triggers error messages when there should be none. The number of false positives in this category may lessen when one increases the number of relaxed constraints, thus permitting more complete analyses. One should probably also use a confidence score for the result of the diagnosis which would be lowered for incomplete analyses.

The 'others' category contains the false positives which could not be classified in the four preceding categories.

In order to reduce the number of false positives, we have proposed some ideas in the previous paragraphs. Implementation of these ideas, however, entail much minute work on the parser and the diagnosis tool. It is certainly possible to diminish the number of false positives given in this paper, and therefore improve the overall results, but it is difficult to evaluate how much work on the *Fips* parser it requires.

5.3 Other results

After having observed the result of parsing with the three constraints relaxed at the same time, we have also examined the results of the six parses when only one or two constraints are relaxed together. In all cases, and especially for errors in number and person, there is a decrease in recall. This is due to the fact that some errors concern two or three constraints together. If one constraint is not relaxed, there cannot be any combination of structures, and errors on the other constraints are not discovered. On the other hand, precision increases somewhat, unfortunately not enough to become satisfactory. Moreover, if constraints are relaxed separately from one another, each sentence has to be parsed several times by the parser, once for each relaxed constraint, which would be time consuming.

Complexity and speed of the system are also of concern to us. Indeed, if the diagnosis takes too long, some users may become bored and prefer not to use the diagnosis system any more. We compared the time necessary to parse the corpus when one, two, or three constraints are relaxed at the same time. Results, shown in Table 3, are relative to the time needed when there is no constraint relaxation.

As expected, parsing time increases with the number of relaxed constraints. Indeed, the number of alternatives which must be taken into account during parsing increases when there are more relaxed constraints. One must notice, however, that it is faster to parse the corpus only once with the three constraints relaxed at the same time than to parse it three times with only one constraint relaxed at a time. Given this result and the previous ones, it seems more interesting to relax all the constraints together than one by one.

6 Concluding remarks

Constraint relaxation can certainly be used for error diagnosis for some categories of the errors which need to be treated. One must determine with precision which are those

Table 3. *Increase in parsing time depending on the number of relaxed constraints*

Number of relaxed constraints	0	1	2	3
Increase in time	—	4,8%	13,2%	23,2%

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

categories, with the help of corpora analyses done by the CECL, and find alternatives for the remaining categories. Several options must be investigated. Partial analyses reconstruction seems an interesting idea, as long as one can guarantee that the partial analyses returned are coherent. We will probably need to find new heuristics to select partial analyses among all those computed by the *Fips* parser. Revising the way scores are given might be a way to select better alternatives, for partial as well as complete analyses. Weighting of the scores linked to the type of errors detected would also be useful.

We are still at the beginning of our research, but we already know that we have to be particularly attentive to false positives and that we need to reduce their number in order to reach as high a precision as possible. Tschichold (1999) says that one needs to strive for a 100% precision, but unfortunately she does not indicate the precision threshold necessary to use a diagnosis system in CALL software. We also need to relax constraints that are less dependent on one another than agreement constraints to check the results presented in this paper on a larger scale. It is possible that the relaxation of other types of constraints does not have the same effect on parsing time or on false positives. Parsing time so far seems satisfactorily short. On average, a sentence is diagnosed in less than half a second. Naturally, one must take into account that the diagnosis tool will be integrated in the CALL software and that some time is needed for the CALL modules to transform the diagnosis output into relevant feedback for the learners. One should also note that the corpus we used for the experiment does not correspond exactly to part of the input of the diagnosis system we are expecting. In our CALL software, input of the diagnosis system will partly consist of isolated sentences as answers to exercise questions, rather than full paragraphs. Isolated sentences may be shorter and their structure perhaps simpler than those found in the corpus. This will ease the task of the parser and may help us to obtain better results in the final system.

Acknowledgements

Deepest thanks to Sylvaine Granger and her team at *Université Catholique de Louvain* for allowing me to use the FRIDA corpus and to the anonymous reviewers for their constructive comments

References

Abney, S. (1987) *The English Noun Phrase in its Sentential Aspect*, PhD Dissertation, Cambridge, Mass: MIT Press.

Chomsky, N. (1981) *Lectures on Government and Binding*, Dordrecht: Foris Publications.

Etchegoyhen, T. and Wehrli, E. (1998) Traduction automatique et structures d'interface. In: *TALN'98*: *Proceedings*, Paris, 2–11.

Felshin, S. (1995) The Athena Language Learning Project NLP System: A multilingual System for Conversation-based Language Learning. In: Holland, V. M., Kaplan, J. D. and Sams, M. R. (eds.), *Intelligent Language Tutors: Theory Shaping Technology*, Hove, UK: Erlbaum, 257–272.

Gaudinat, A., Goldman, J.-P. and Wehrli, E. (1998) Le système de synthèse FIPSVox: syntaxe, phonétisation et prosodie. X*XIIth JEP: Proceedings*, Martigny, Switzerland, 139–142.

Granger, S. and Meunier, F. (1994) Towards a grammar checker for learners of English. In: Fries,

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press

U., Tottie, G. and Schneider, P. (eds.), *Creating and Using English Language Corpora*, Amsterdam, Atlanta: Rodopi, 79–91.

Haegeman, L. (1994) *Introduction to Government and Binding Theory*, Oxford, UK; Cambridge, USA: Blackwell.

Hamel, M.-J. and Vandeventer, A. (2000) Adapter un analyseur syntaxique et l'intégrer dans un système d'ELAO: le cas de FIPSGram dans SAFRAN. In: Laurier, M. and Duquette, L. (eds.), *Enseignement-apprentissage de la langue seconde dans des environnements multimédias*, Montreal: Editions Logique, 117–136.

Heinecke, J., Kunze, J., Menzel, W. and Schröder, I. (1998) Eliminative Parsing with Graded Constraints. *Coling-ACL'98:Proceedings* **1**, 526–530.

Kübler, N. and Cornu, E. (1994) Using automata to detect and correct errors in the written English of French-speakers. *TRANEL*, **21**: 235–245.

Laenzlinger, C. and Wehrli, E. (1991) *FIPS*: un analyseur interactif pour le français. *TA Information*, **32**(2): 35–49.

Matthews, C. (1993) Grammar Frameworks in Intelligent CALL. *CALICO*, **11**: 5–27.

Murphy, M., Krüger, A. and Grieszl, A. (1998) RECALL – Providing an Individualized CALL Environment. In: Jager, S., Nerbonne, J. A. and van Essen, A. J. (eds.), *Language Teaching and Language Technology*, Lisse: Swets & Zeitlinger, 62–73.

Schneider, D. and McCoy, K. F. (1998) Recognizing Syntactic Errors in the Writing of Second Language Learners. *Coling-ACL'98: Proceedings 2*, 1198–1204.

Tschichold, C. (1999) Grammar checking for CALL: Strategies for improving foreign language grammar checkers. In: Cameron, K. (ed.), *CALL: Media, Design & Applications*, Lisse: Swets & Zeitlinger, 203–22.

Tschumi, C., Bodmer, F., Cornu, E., Grosjean, F., Kübler, N. and Tschichold, C. (1994) The ARCTA prototype: An English writing tool and grammar checker for French-speakers. *TRANEL*, **21**: 223–228.

Vandeventer, A. (1999) FIPSGram: A Tool to Create Sentence Structure Representations for CALL. In: Zinglé, H. (ed.), *JILA'99: Proceedings*, Nice: Travaux du LILLA, 272–275

Wehrli, E. (1997) *L'analyse syntaxique des langues naturelles: problèmes et méthodes*, Masson: Paris.

Weischedel, R. M. and Black, J. E. (1980) Responding Intelligently to Unparsable Inputs. *American Journal of Computational Linguistics*, **6**(2): 97–109.

https://doi.org/10.1017/S095834400100101X Published online by Cambridge University Press