ReCALL **12** (1): 79–91. Printed in the United Kingdom

© 2000 Cambridge University Press

Reusing a syntactic generator for CALL purposes

ANNE VANDEVENTER
LATL, Université de Genève, Switzerland
(*e-mail: vandeventer@latl.unige.ch*)

MARIE-JOSÉE HAMEL
UMIST, Manchester, UK

Abstract

This article presents briefly the advantages and disadvantages of reusing natural language processing (NLP) tools in the CALL context. The issue is addressed through the description of GBGen, a sentence generation system. The abstract semantic representation used as input for the generator is described, as well as the actual generation process, from a deep structure to a grammatical sentence through transformations and the application of morphology. The possible didactic value of such a tool is then evaluated and the outline of a CALL scenario given. Finally, proposed adaptations of the generator for the CALL context are discussed.

1 Introduction

For some time now we have looked into the possibility of reusing NLP components for CALL purposes, by adapting them for a learning clientèle and integrating them into a CALL environment where they would be used to support various learning tasks. Previous work (Hamel and Vandeventer, 1998; Hamel, 1998, 1997) made suggestions regarding the potential reusability of two NLP components: FIPSGram (Vandeventer, 1999), a sentence structure viewer, and FIPSVox (Gaudinat and Goldman, 1998; Gaudinat and Wehrli, 1997), a speech synthesiser, within the prototype of a CALL environment called SAFRAN[1], *Système pour l'Apprentissage du FRANçais*, which targets the intermediate to advanced adult learner population. The present paper reports on work in progress exploring the benefits of reusing GBGen[2] (Etchegoyhen *et al.*, 1999; Etchegoyhen and Wehrle, 1998), a sentence generator, for similar CALL purposes.

The next section addresses the problem of reusability in CALL. Section 3 describes the generator GBGen. Section 4 shows GBGen's didactic value. Section 5 reviews adaptations to be made on GBGen before using it for CALL. Section 6 offers some concluding remarks.

[1] The SAFRAN project was subsidised by the FRANCIL network of the AUPELF-UREF, n° X/1.20.09.01.1/97.07.0
[2] The GBGen project was supported by grant n° 12-50797.97 from the Swiss National Science Foundation.

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

2 Reusing NLP tools

The issue of reusing already existing NLP tools in CALL as opposed to building them from scratch has not, to our knowledge, been often addressed in the CALL literature. Indeed, apart from mentioning that NLP is actually 'hard' and rarely goes beyond the syntactic level, and that NLP-based CALL software never goes beyond the prototype stage (Holland *et al.*, 1993), authors have rarely compared the benefits and the problems linked with the development of NLP-based CALL systems using new vs. borrowed NLP components. In most known cases, the NLP components used are actually new ones. Previous experiences in CALL shows that these dedicated NLP components are of very limited capacity because of their lack of 'maturity', i.e. robustness, rapidity, reliability, etc. We therefore looked into the possibility of reusing already existing, more mature NLP tools. However, the task of finding reusable NLP components is not straightforward and presents some difficulties worth mentioning.

If the question of reusability has not been discussed within the field of CALL as such, it has certainly been raised as an important issue within the field of computer science (Biggerstaff and Perlis, 1989; Prieto-Diaz, 1993) and, more specifically, within the field of computational linguistics. Cunningham *et al.* (1997), among others, mentions general problems associated with the notion of software reuse, namely:

1. The overall architecture of software might be too domain-specific for one or more of its components to be reusable.
2. Some software components might also be too granular i.e. 'designed at system level' (Cunningham *et al.*, 1997: 358), therefore not reusable.
3. Designing components that are 'specific enough to be useful in a domain' (Cunningham *et al.*, 1997) but generic enough to be reusable costs much more than designing non-reusable components (a 4:1 proportion).
4. The lack of documentation and results accompanying the development of software components to be reused constitutes an important problem.
5. Finally, the difficulty of transferring software components to another platform and/or into another programming language might make them ultimately non-reusable.

To these problems generally associated with the notion of software reuse must be added the problem of availability that arises when looking more specifically for NLP components such as taggers, parsers, or speech synthesisers, which are often designed as language specific tools. Therefore, finding reusable NLP components for a given language, especially if this language is not English, can be a hard or even impossible mission.

The NLP tools developed at the Laboratoire d'Analyse et de Technologie du Langage (LATL) are likely candidates for reuse because they are modular, non-specific, transferable, and available, while offering a relatively large and reliable coverage of the French language. However, reuse has its costs and one should not underestimate the time necessary to transfer NLP components to an interface suitable for CALL purposes. The individual adaptation of the NLP components, which is the making of a 'comprehensible' (Swain, 1985) output, has led us over the past years to some successful outcomes. Finally, the integration of the adapted NLP tools within the CALL environment, which implies anticipating when, how and why the NLP components will work within the overall CALL scenario, is currently being explored.

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press



Fig. 1. From a pseudo-semantic structure to a sentence.

3 Description of GBGen

GBGen is a syntactic generation system, based on the Government and Binding framework (see among others Chomsky, 1981, 1986; Haegeman, 1994). GBGen was developed for French, but most of its procedures are language independent, thus portable to other languages. Its input is a relatively abstract representation of the semantics of the sentence to be generated, called Pseudo-Semantic Structure (PSS). The generation process follows three steps:

1. from the PSS a deep structure (DS) is created,
2. transformations and movements of elements of the DS result in a surface structure (SS),
3. morphology converts the SS into a grammatical sentence (see Figure 1).

We will now illustrate the conversion of a pseudo-semantic structure into a grammatical sentence through deep structure, surface structure and the application of morphology.

3.1 Pseudo-Semantic Structures (PSS)

The semantic representation used as input for GBGen is a pseudo-semantic structure (PSS) (see Etchegoyhen *et al.*, 1999; Etchegoyhen and Wehrle, 1998; Etchegoyhen, 1997). A PSS is a generic object containing abstract information (e.g. thematic role) and pointers to a list of related PSS (the satellites), to single other PSS (e.g. co-ordination), or to one of the three PSS subtypes. These three subtypes are

(i) clause structure (CLS), containing a predicate, usually a lexical verb, as well as functional information such as tense, voice and type of utterance;
(ii) determiner structure (DPS) containing a lexical property (a noun) associated to an abstract operator (c.f. Barwise and Cooper, 1981) and agreement features;
(iii) characteristic structure (CHS) used for modifiers such as adjectives and adverbs.

Let us take a concrete example with the sentence in 1(a) and the corresponding simplified PSS 1(b) (only the fields containing non-null information are displayed).

1(a) *C'est ses expériences que le narrateur raconte*
It is his experiences that the narrator tells

1(b) (see Figure 2)

The semantics of the sentence and of the top PSS are closely linked. However, the shape of the final sentence may not be obvious after a quick look at the PSS. Indeed, in 1(b), there seems to be only one clause, with *raconter* as its main verb, while in 1(a) there are

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

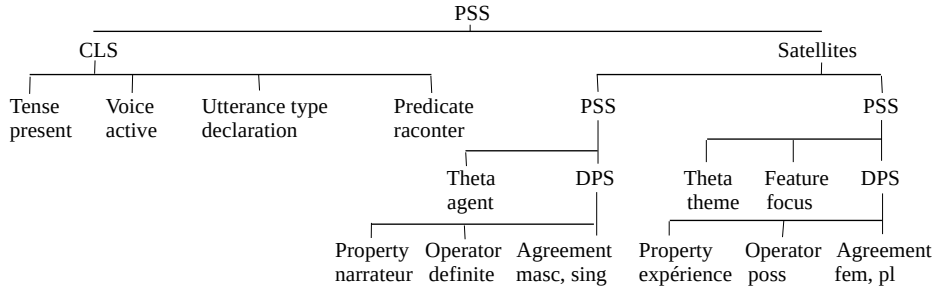


Fig. 2. Example 1(b) (see text).

two clauses, the one containing *raconter* embedded in the other. But for the feature 'focus' carried by the embedded PSS bearing the thematic role (Theta in 1(b)) 'Theme', the PSS in 1(b) would generate (2):

2. *Le narrateur raconte ses expériences*
The narrator tells his experiences

The CLS embedded in the outer PSS contains tense and voice information (active), as well as the type of utterance (declarative). The predicate, the verb *raconter*, is mentioned. The outer PSS contains a list of two satellites, each of which bears the thematic role of its embedded DPS. As mentioned above, the 'Theme' PSS also bears the 'focus' feature which will be used to generate the cleft structure. The two DPS are rather similar, containing both a property on which an operator acts, and agreement features.

3.2 Deep Structure (DS)

A Deep Structure (DS) is generated from the PSS. The DS of a sentence contains its elements in their base positions. To create a DS, one starts with the predicate of the highest CLS in the PSS structure, usually a verb. On the basis of this verb, a full verb phrase (VP) associated to a tense phrase (TP) and a complementizer phrase (CP) is generated, giving the representation in (3)[3]:

3. $[_{CP}\ [_{TP}\ [_{T'}\ [_{VP}\ ]]]]$

A search in the lexicon is initiated to retrieve the lexeme corresponding best to the predicate of the CLS and the associated satellites. The verb is then placed inside the VP, and tense information is attached to it. In the case of *raconter*, the argument list retrieved from the lexicon indicates that it takes two arguments, an 'Agent' as subject and a 'Theme' as object. The 'Agent' DPS is then placed in subject position, that is in the

[3]Bracketed representations are used for DS and SS, showing the elements, their categories, and the relationships between the different elements.

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

specifier of the TP, and the 'Theme' as the complement of the VP. In this case, both DPS are realised as determiner phrases (DP, see Abney, 1987) as the verb does not select prepositional phrases. The first DPS is rendered as the definite article followed by the noun, and the second by a possessive adjective (poss) followed by the noun. Because of the focus feature, the latter DP is marked as a cleft element. This gives the representation in (4), which is the same representation as the DS for (2). However, some features do not appear in the bracketed representation, especially cleft information, tense, etc.

4. $[_{CP}\ [_{TP}[_{DP}$ le [NP narrateur]]$[_{T'}\ [_{VP}$ raconter $[_{DP}$ poss $[_{NP}$ expérience]]]]]]

3.3 Surface structure

From the DS, the surface structure (SS) is reached through the movement of some elements, sometimes triggering the creation of additional structure. The transformation algorithm checks each element for features requiring its movement, such as tense, cleft marks, and pronouns in unlicensed positions. In the case of (4), two elements will undergo movement: the verb and the object. The verb *raconter* must move from the head of the VP to the head of the tense phrase (TP) because it is tensed. The object of *raconter*, [DP poss [NP *expérience*]], is marked as a cleft element and must occur in focus position. When an element bearing the cleft feature is encountered, a new structure such as (3) is generated, with ce as subject and the copula as its main verb. Tense features from the main verb in the DS are copied into the copula. The intermediate structure resulting from both the DS and movements before clefting is inserted as the complement of the new structure's VP. Finally, the projection which triggered the cleft transformation is moved from its base position to the specifier of the embedded CP, leaving a trace, as shown in (5).

5. $[_{CP}\ [_{TP}[_{DP}$ ce $]][_{T'}$ être $[_{VP}\ [_{CP}[_{DP}$ poss $[_{NP}$ expérience $]]_i\ [_{C'}\ [_{TP}[_{DP}$ le $[_{NP}$ narateur]] $[_{T'}$ raconter $[_{VP}\ [_{DP}$ ei]]]]]]]]]

At this stage, the structure is complete and all the elements present are in the appropriate position. Some elements are still missing or not inflected, in particular verbs and pronouns, as their morphological shape strongly depends upon their SS position. Morphology takes care of these last changes.

3.4 Morphology

The role of morphology in GBGen is twofold. It must both find the correct inflection for all the words of the sentence, and lexicalise some elements such as pronouns and complementizers.

For each word of the sentence, the lexicon is accessed and the word form corresponding to the agreement information is retrieved. Ellipses and contractions are also performed at this stage.

After morphological procedures apply to (5), one obtains the structure 6(a), with the full grammatical sentence. For readability's sake, the system provides a version 6(b) without the syntactic structure representation.

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

6a. $[_{CP}\ [_{TP}[_{DP}$ c' $][_{T'}$ est $[_{VP}\ [_{CP}[_{DP}$ ses $[_{NP}$ expériences $]]_i$
$[_{C'}$ que $[_{TP}[_{DP}$ le $[_{NP}$ narrateur $]]$
$[_{T'}$ raconte $[_{VP}\ [_{DP}\ e_i\ ]]]]]]]]]$

6b. C'est ses expériences que le narrateur raconte

4 Potential didactic value of GBGen

GBGen in its actual format is a far too complex NLP tool to be used as such in CALL software. However, behind its complexity, GBGen contains a didactic value which could be exploited given the proper CALL environment and scenario. GBGen's didactic value comes essentially from the fact that its operation is user-dependent, i.e. the sentence generation process can only be completed if some information is provided to the system by the users. This makes the users essential partners in the sentence generation process, and the quality of the resulting output therefore depends not exclusively on the system's own performance, but also on the relevance of the information provided by the users. This partnership, which most obviously benefits the machine, can also benefit the users, especially if they are language learners, essentially because of the active role they are requested to play in the sentence generation process. Indeed, GBGen could support the language learners in:

1. developing their autonomy,
2. focusing on both production and comprehension aspects of the language,
3. noticing 'gaps' in their target language system, and
4. promoting self-correction strategies.

4.1 Developing learners' autonomy

GBGen cannot function without user input; moreover, the input provided must be adequate in order to guarantee the quality of the resulting output. This forces users to act as 'experts' in the sentence generation process, hence giving them a high level of responsibility. In order to fulfil this role for the system, the users will have to take the initiative of calling upon their own resources and ideally also the resources available throughout the CALL environment, such as tutorials, dictionaries, reference grammar, other NLP tools, etc.

4.2 Focusing on both production and comprehension aspects of the language

GBGen uses the information provided by the users to construct sentences 'automatically'. Sentences are therefore not created by accident. All the elements which constitute the sentences will have been anticipated by users who already have either a whole sentence pattern in mind which they want the system to recreate, or some isolated ideas of sentence units from which they wish the system to generate a sentence.

Learners are hence invited to focus on the production aspect of the target language when they are asked, at the beginning of the sentence generation process, to think of a sentence of their own choice. This sentence will be stored and compared at the end of

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

the generation process with GBGen's output, arising directly from the information given by the learners. The grammatical coverage of GBGen being large enough, the learners will not be restricted in their choices, allowing more room for creativity and therefore authentic production activities.

The learners are also invited to focus on the comprehension aspect of the target language. Whether a sentence is being wholly anticipated or not by the learners, the information needed to enable the 'automatic' generation forces them to think in terms of its 'smaller units' i.e. to provide an analysis of the sentence's internal structure. This form of comprehension activity, because of its high level of abstraction, can be a demanding task for language learners; thus, provision must be made so that the CALL environment resources mentioned above are available to them during this task.

4.3 Noticing 'gaps' in learners' target language system

Requesting the learners to use their analytic skills will raise their grammatical consciousness which in turn is known to facilitate noticing (Schmidt, 1990), a conscious process by which the learners are made aware of 'gaps' in their own target language system. Noticing can occur with or without the help of external aids (feedback). Swain (1985, 1998), for instance, suggests that feedback helps by bringing the learners' attention to 'a linguistic problem' in their own output. Noticing, however, could also happen when the learners, facing difficulties while trying to produce an output, notice 'gaps' in their own target language system by themselves, i.e. without having access to any form of feedback. With GBGen, the learners might for instance discover that they can effectively think of sentences of their own as an entity – and sometimes fairly complex ones – without being really able to understand the details of the whole sentence's overall/internal structure. It is hoped that GBGen, by actually forcing the learners to de/re-construct sentences into/from their smaller units, will allow them to go beyond some of the linguistic gaps they might have noticed in their linguistic system, using their own resources and the resources available within the CALL environment.

4.4 Promoting self-correction strategies

With GBGen's current format, the feedback provided to the learners is the actual output, the sentence generated by the system. This output is precious as the learners can use it to compare it with their original sentence and correct themselves, if necessary. However, a mismatch between the two sentences does not automatically mean that the learners' original sentence was incorrect: it could mean that some of the information provided by the learners was inadequate. In order to know which diagnosis applies, the learners have to review the information they provided. The most obvious place to start (not necessarily the easiest one for the learners) would be to look at the sentence bracketed output to check for discrepancies in the structure generated by GBGen. Then, they could look back at the information they have given to the system during the generation process. To do this, the learners need access to the information already given and some form of assistance in evaluating the relevance of the information. In the next section, we make suggestions as to how GBGen can best adapt to the needs of a language learning clientèle.

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

5 Adaptations of GBGen

Given the complexity of the pseudo-semantic structures, several adaptations are necessary before GBGen can be usefully integrated into a CALL environment. Seven types of adaptation are envisaged and discussed below. Three relate to simplifications of terminology and options:

(i) new terminology and labels,
(ii) reducing the number of options,
(iii) offering default options.

One concerns the output:

(iv) output simplification.

The last three pertain to the way the information is requested and to the coherence of that information:

(v) PSS checker,
(vi) reordering the information requests,
(vii) interactive dialogues.

5.1 New terminology and labels

Language learners and teachers do not know the pseudo-semantic terminology and should not need to learn it. Thus, they should be provided with sentences and paraphrases explaining the kind of information expected from them at any given time, in the target language. For all the PSS terminology to be eliminated, appropriate labels for the numerous values that GBGen's options can take must be found, providing clear and precise information on the desired forms, while retaining some level of abstractness.

Some examples of GBGen original labels and the corresponding new ones are given in Table 1. When the mouse is located on labels, the paraphrases in quotes are shown. On double-click, additional information about the label is provided and examples in context given.

Table 1. *Examples of GBgen original labels and corresponding new ones*

GBGen label	New label	Paraphrases
PSS	PH	*'Information sur la structure générale de votre PHRASE'*
CLS	PRED	*'Information sur le PRÉDICAT'*
theta	ROLE	*'ROLE de l'argument'*
DPS	SYNG	*'Information sur le contenu du SYNTAGME'*
CHS	MOD	*'MODIFIEURS du prédicat/du syntagme'*
Utterance type	TYPE	*'TYPE de phrase'*
Feature	MARQ	*'Phrase MARQUÉE'*

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

Table 2. *Mapping of GBGen's mode and aspect into French tenses*

Mode		*Aspect*			
Colour	*Tense*	*- progressive - perfective*	*- progressive + perfective*	*+ progressive - perfective*	*+ progressive + perfective*
Real	E<S	Ind. passé simple	Ind. plus-que-parfait	Ind. imparfait	Ind. plus-que-parfait
	E=S	Ind. présent	Ind. passé composé	Ind. présent	Ind. passé composé
	S<E	Ind. futur	Ind. futur antérieur	Ind. futur	Ind. futur antérieur
	N/A	Infinitif	Infinitif	Part. présent	Part. passé
Unreal	E<S				
	E=S	Subj. présent	Subj. passé	Subj présent	Subj. passé
	S<E	Cond. présent	Cond. passé	Cond. présent	Cond. passé
	N/A	Subj présent	Subj. passé	Subj. présent	Subj passé.
N/A	E<S	Infinitif	Infinitif passé	Infinitif	Infinitif
	E=S				
	S<E				
	N/A				

5.2 Reducing the number of options

Sometimes, options are too numerous to be presented all at once to the users, and several options combine to give only one value. Verb tenses are a typical example of the combination of options. In GBGen, tenses are derived from the combination of aspect and mode. Aspect regroups perfective and progressive information, while mode regroups colour and tense. Tense is represented by the position on a time scale of the event (E) and the speaking (S) point ('E < S' represents past, 'E = S' present, and 'S < E' future). To spare the learners all the possible combinations, GBGen variables are mapped into a table representing the different French tenses and modes (see Table 2). Only tense and mode need to be provided.

5.3 Offering default options

Some options can be set by default. For example, a PSS containing a characteristic structure (CHS) always has the value 'set restriction'. Choosing the 'pronoun' option for a DPS automatically sets the property and operator fields to empty. Gender, number and person for a DPS are pre-set to masculine, singular, third person. Users must be able to modify default options if they so desire, the goal of default options being to help them in their task and not to make definitive choices for them.

5.4 Output simplification

The output of GBGen, as we have seen in (6), is both the grammatical sentence 6(b) and

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

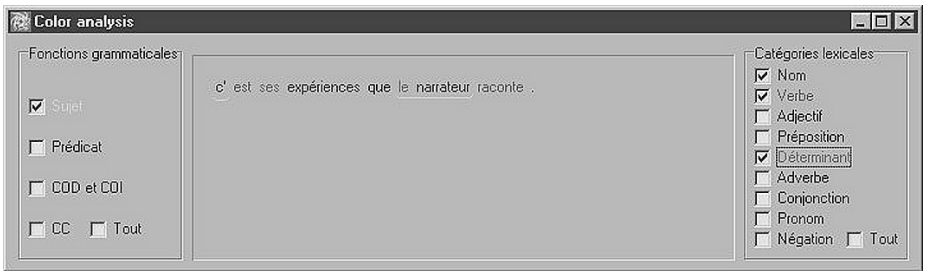


Fig. 3. FIPSGram colour coding.

the sentence structure 6(a), a structure designed for generative linguists and hard to decipher for French learners. In order to give the learners a simplified and easier to understand version of the structure of their sentences, a sentence structure viewer called FIPSGram (see Hamel and Vandeventer, 1998; Vandeventer, 1999) is used. FIPSGram provides graphical representations of the structure of grammatical French sentences through either a tree representation, or colour coding of the lexical categories and grammatical functions, as shown in figure 3.

5.5 PSS checker

Because PSSs will be created interactively and because the users will be mostly learners of French, it is necessary to verify the correctness of the PSS in order to guarantee a grammatical output from the generator. This checking and diagnostic module will be able to inform the users if they have not employed the correct number or types of arguments, if they declared their sentence to contain a cleft but did not indicate on which element the cleft occurred, if they indicated the wrong gender for a noun, and similar discrepancies.

5.6 Reordering the information requests

The pieces of information necessary to create a complete PSS will not be requested from the users in the order they are displayed in the PSS. The learners must first think of the general shape of the sentence they want to create. They have to decide whether their sentence is complex or not, containing perhaps an embedded sentence such as a relative or a cleft structure. In the PSS, this type of information is not visible at the higher levels, but attached to the modified or focused element. In the dialogue leading to the creation of a PSS, information changing the overall shape of the sentence (e.g. MARQ, *'Phrase marquée'* for a cleft) is requested very early on. This information will be presented to the learners at the level of the sentence type and of the predicate, rather than embedded deep down in the structure.

5.7 Interactive dialogues

Interactive dialogues will lead the learners through the different steps needed for the

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

creation of the PSS. Most of the requests in these dialogues will be presented as interrogative sentences and answered through the use of pull-down menus, check boxes, or a word to be typed in.

For the sake of clarity, let us go through a few steps of such a PSS creation of dialogues in English (keeping in mind that such dialogues would be in French in the envisaged French CALL system).

Step one: a sentence in mind?

Dialogue Box N° 1

Context: The learner chooses between the option of anticipating or not a sentence in advance.

A. Do you want to think of your own sentence before starting to work with GBGen (yes/no)
B. If yes: You can write down/say your sentence here:

Dialogue Box N° 2 (optional)

Context: If the learners have been previously identified as new users of the system, they will be offered the option of seeing the structure of their anticipated sentence with FIPSGram. This should provide them with some useful clues for the next step.

A. Would you like to see your sentence with FIPSGram? (yes/no)

Step two: you're the expert!

Dialogue Box N° 3: PH: 'Overall aspect of my sentence'

Context: The learner is asked to decide on the overall aspect of the sentence to generate.

A. What TYPE of sentence do you want GBGen to generate: affirmative, negative, interrogative, imperative? (pull-down)
B. Will this sentence be MARKed?: by a passive, by a cleft, ...? (pull-down)

Dialogue Box N° 4: PRED: 'Information about the predicate of my sentence'

Context: The learner is asked to provide information about the predicate of the sentence.

A. What is the NATURE of sentence's predicate: verb, adjective, etc.? (pull-down)
B. NAME/DECLARE the predicate here:
C. If the predicate you have in mind is a verb, then:
   – What is its MODE?: (pull-down)
   – What is its TENSE?: (pull-down)

Dialogue Box N° 5: SAT: 'Satellites for my predicate'

Context: The learner is asked to declare the number of arguments and adjuncts 'surrounding' the predicate.

A. How many arguments does your predicate have? (numeric answer)
B. Do you want to add some adjuncts to your sentence? (yes/no)
   If yes, how many? (numeric answer)

Dialogue Box N° 6: ARG: 'Information about each argument of my predicate'

Context: The learner is asked to provide specific information about each argument of the predicate.

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

A. NAME/DECLARE the argument here:
B. What is the ROLE of the argument in the sentence: agent, patient, theme, ... (pull-down)
C. Is the argument a CLAUSE or a PHRASE? (check box)
D. If the argument is a phrase, what kind of phrase? nominal, verbal, etc. (pull-down)

At any time during the sentence generation process, learners can ask for more information about the labels used in the dialogues by double-clicking on them. An on-line grammar will provide users with explanations and examples in context.

The adaptations described in this section should enable the GBGen system to become a useful tool for CALL, by making it accessible to non-linguists and by providing it with a user-friendly interface. However, work remains to be done on CALL environments and scenarios for GBGen's richness to be exploited in the best possible ways.

6 Conclusions

GBGen is a specialised tool geared towards the syntactic level of sentence creation. As such, it is de-contextualised and each sentence will be treated individually. This tool will not help the learners to decide what to say or how to build a coherent paragraph. However, on the syntactic level, GBGen covers most constructions of French and should be able to generate most intended sentences. Then, given the envisaged CALL environment, the users will ultimately learn how these different sentences are used in French within different types of texts.

GBGen is designed to be as language independent as possible, using language-generic algorithms whenever possible and resorting to language-specific procedures mostly for morphology. Thus, it is easily portable to other languages. GBGen is also reusable in its end applications. GBGen was not initially intended to be used in a CALL context but its adaptation to language learning requires no change of the pseudo-semantic structures or the generation process itself, just a new user interface.

As stated in the introduction, this article is reporting on some work in progress. Although the GBGen system is operational, the realisation of its CALL interface has just started. It is hoped that the length of time necessary to complete a full PSS through the set of questions mentioned in 5.7 will not discourage learners to use the system. A user-friendly interface, stimulating dialogues and explicit choices of options should ease this process.

Acknowledgement

The authors would like to thank Dr. Catherine Walther Green for her suggestions and comments on earlier versions of this paper.

References

Abney, S. (1987) The English Noun Phrase in its Sentential Aspect. PhD dissertation, Cambridge: MIT.

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press

Barwise, J. and Cooper, R. (1981) Generalized quantifiers and natural languages. *Linguistics and Philosophy*, **4**: 159–219.

Biggerstaff, T. J. and Perlis, A. J. (1989) *Software Reusability.* ACM Press/Addison-Wesley.

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

Chomsky, N. (1986) *Barriers*. Cambridge: MIT Press.

Cunningham, H., Freeman, M. and Black, W. J. (1997) Software reuse, object-oriented frameworks and NLP. In: Jones, D. and Somers, H. L. (eds.), *New Methods in Language Processing*. London: UCL Press, pp. 357–366.

Etchegoyhen, T. (1997) Génération automatique de phrases: le système GBGen. Mémoire de Diplôme d'Etudes Supérieures. Université de Genève, Geneva: Manuscript.

Etchegoyhen, T. and Wehrle, T. (1998) Overview of GBGen: A large scale domain independent syntactic generator. *9th International Workshop on Natural Language Generation: Proceedings*, Niagara-on-the-Lake, pp. 288–291.

Etchegoyhen, T., Mengon, J., Vandeventer, A. and Wehrle, T. (1999) Une approche efficace à la génération syntaxique multilingue: Le système GBGen. *GAT'99: Proceedings*, Grenoble: GRESEC, Université Stendhal-Grenoble 3, pp. 135–144.

Gaudinat, A. and Goldman, J.-P. (1998) Le système de synthèse FIPSVox: syntaxe, phonétisation et prosodie. *XXIIèmes Journées d'Etudes sur la Parole: Proceedings*, Martigny: IDIAP, pp. 139–142.

Gaudinat, A. and Wehrli, E. (1997) Analyse syntaxique et synthèse de la parole: le projet FIPSVox. *TA-Informations*, **38**(1): 121–134.

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

Hamel, M.-J. (1998) Les outils de TALN dans SAFRAN. *ReCALL*, **10**(1): 79–85.

Hamel, M.-J. (1997) NLP tools in CALL for error analysis. *Journal of the CAAL*, **18**(2): 125–142.

Hamel, M.-J. and Vandeventer, A. (1998) FIPS-Gram: Un analyseur syntaxique dans un contexte d'ELAO. *Proceedings NLP+IA'98 Conference*. Moncton: Université de Moncton, pp. 18–24.

Holland, M., Maisano, R., Alderks, C. and Martin, J. (1993). Parsers in tutors: What are they good for? *CALICO*, **11**(1): 28–45.

Prieto-Diaz, R. (1993) Status report: Software reusability. *IEEE Software*, **10**: 61–68.

Schmidt, R. (1990) The role of consciousness in second language learning. *Applied Linguistics*, **11**(2): 17-46.

Swain, M. (1985) Communicative competence: Some role of comprehensible input and comprehensible output in its development. In: Gass, S. and Madden, C. (eds.), *Input in Second Language Acquisition*. Rowley, MA: Newbury House, pp. 235–253.

Swain, M. (1998) Focus on form through conscious reflection. In: Doughty, C. and Williams, J. (eds.), *Focus on Form in Classroom Second Language Acquisition*. Cambridge: CUP, pp. 64–82.

Vandeventer, A. (1999) FIPSGram: A tool to create sentence structure representations for CALL. In: Zinglé, H. (ed.), *JILA'99: Proceedings*, Nice: Publications de la faculté des Lettres Arts et Sciences humaines de l'Université de Nice-Sophia Antipolis, pp. 272–275.

Anne Vandeventer is a research and teaching assistant at the Laboratoire d'Analyse et de Technologie du Langage, Department of Linguistics, University of Geneva. She is currently working towards her doctorate, specialising in the use of NLP tools in CALL, and more particularly in grammar checkers for language learners.

Marie-Josée Hamel is a lecturer at the Centre for Computational Linguistics, Department of Language Engineering, UMIST. She teaches French as a foreign language, translation and CALL theory. Her research interests include the use of NLP tools in CALL and the design of CALL environments that draw upon SLA theory.

https://doi.org/10.1017/S0958344000000914 Published online by Cambridge University Press