THE TEACHER MODE OF THE SENTENCE FAIRY SYSTEM: HOW TO CREATE YOUR OWN E-LEARNING WRITING LESSONS FOR GERMAN ELEMENTARY SCHOOL PUPILS

Karin Harbusch, Christine Franz, Ulrich Koch

University of Koblenz-Landau, Computer Science Dept. (GERMANY)
harbusch@uni-koblenz.de, tinefr@uni-koblenz.de, koch@uni-koblenz.de

Abstract

The Sentence Fairy system implements a “virtual writing conference” for elementary-school children, with German as the target language. It deploys natural-language generation technology to evaluate and improve the grammatical quality of learner output. Based on an abstract representation of the story under construction, all paraphrases of simple and combined clauses are generated fully automatically. From this source, the system produces exercises enabling the learners to improve their sentences. So far, the abstract story representations had to be written by skilled linguists. Through the new teacher mode, users with elementary linguistic knowledge about constituents of simple clauses can enter their own story, which is then automatically translated into e-learning exercises. The teachers are supposed to fill constituents of simple clauses into a drop-down menu, with the system adding grammatical functions such as Subject, Direct Object, Modifier, etc. This way of specifying a sentence prevents syntactic ambiguity, thereby promoting the accuracy of feedback generation by the system. The information on rhetorical relations such as causality and temporal order*,* which provide the basis for a variety of sentence combination exercises, is also entered in a dialogue without technical terms. Moreover, the teacher mode allows the specification of lexical-choice alternatives. The system automatically generates a suggestion list for all words of the story by selecting synonyms, hyponyms and hyperonyms in WORDNET.

Keywords: ICALL, natural-language processing (NLP), paraphrase generation, e-learning, computer-supported learning, tutoring system, virtual writing conference, sentence combining, lexical choice.

1 MOTIVATION

Imagine the following teaching scenario. You can enrich the writing lessons you give to your elementary school class, with *sentence combining* or *lexical choice* exercises to be performed by each pupil individually, at their own speed, in an e-learning system. The *Sentence Fairy* system gives exact grammatical feedback to any construction used by a pupil, in a manner tailored to their individual proficiency level. During a so-called *writing conference* (a didactic method often used in German class rooms), the class as a whole decides on the preferred improvements for the original (poor) story, based on experiences with the exercises tutored by the Sentence Fairy, and on your own suggestions.

In the following, we describe how the *teacher mode* of the Sentence Fairy system enables the definition of new stories via simple dialogues. The Sentence Fairy system implements a “virtual writing conference” for elementary-school children, with German as the target language. It deploys *natural-language generation technology* to evaluate and improve the grammatical quality of learner output. The Sentence Fairy system can choose between all paraphrases of simple and combined clauses for a teacher-invented story to make up exercises enabling the learners to improve their sentences. The complete list of paraphrases is generated fully automatically by the underlying *COMPASS-II* system. In previous versions of the Sentence Fairy and COMPASS-II systems, abstract story representations could only be devised by teachers trained in formal linguistics. The new teacher mode allows teachers with restricted linguistic knowledge about constituents of simple clauses to enter new stories, which will then be translated into e-learning exercises nearly fully automatically.

Syntactic ambiguity in clauses entered by the teacher has to be avoided. It would often give rise to inexact feedback delivered by the Sentence Fairy system. Thus, the teacher mode restricts *natural language parsing* to constituents instead of whole clauses. The teacher mode provides a drop-down menu with grammatical functions licensed by a chosen verb, i.e. Subject and Direct Object if the Verb is transitive. All grammatical functions provide a mouse-over function showing definitions for the linguistic terms to avoid confusion about names such as Accusative Object vs. Direct Object. Some additional information has to be given as answers to dedicated questions—e.g. Gender of new Proper

Nouns and co-referentiality of identical objects all over the story to create pronominalization and relative clause exercises properly.

The information on *rhetorical relations*, e.g. *causality,* to become the basis for a variety of sentence combination exercises, can be specified by setting marks in a system-generated table providing all sentence pairs. Alternatively, a subordinate conjunction can be typed. The system translates the choice into the abstract relation in *Rhetorical Structures Theory (RST;* e.g. "Peter combed his hair **before** Peter left the house" will become EARLIER(a,b) and can result in "Peter left the house **after** he combed his hair " as an alternative paraphrase generated from the abstract RST relation.

Moreover, the teacher mode allows the specification of *lexical choice* alternatives (e.g. *run, hasten, hustle*). The system automatically generates a *suggestion list* by selecting all synonyms, hyponyms and hyperonyms in WORDNET for all words of the story. As the German variant requires the translation of the German terms to English WORDNET and translating words back again, the teacher can erase wrong translations for unintended meanings of the German term; additionally, s/he can add further alternatives not provided in the suggestion list.

The paper is organized as follows. In the next section, we outline the state of the art in language learning systems based on *natural language processing (NLP)* tools. In Section 3, we sketch the Sentence Fairy system, for which the teacher mode provides input. The output of the teacher mode passes through the paraphrase generator COMPASS-II. In Section 4, we outline the COMPASS-II system, which produces all correct paraphrases that the Sentence Fairy can present as exercises to the pupils. Section 5 describes the three different stages of the teacher mode: First, the teacher enters the story sentence by sentence. Next, s/he indicates the rhetorical relations for the entire story. Section 5.3 deals with the automatic generation of lexical choice variants. In the final section, we draw some conclusions and suggest future work.

2 NLP-BASED ICALL WRITING TOOLS: STATE OF THE ART

Computer-supported learning of how to write grammatically correctly in L1 and L2 figures prominently in the ICALL literature. Here, we cursorily review systems based on *natural-language processing (NLP)* techniques that provide students with online support in improving their writing skills either in L1 or L2. Currently available text manipulation systems typically use prefabricated ("canned") texts. Cloze tests are prepared by the teacher or produced (semi-)automatically by the system. Sentence combining exercises [25] require explicit specification of all possible correct answers. During such an exercise, reliable feedback results from simple (possibly partial) string-matching or more enhanced pattern-matching of student answers with the canned text. A wide variety of such systems is available online on the internet. Students can type their answers into an internet form. These are then compared to the correct answer, thus enabling direct interactive feedback. An example is the system "Fun with Texts*"* by Camsoft [3].

Virtually the entire literature on NLP applications to the syntactic aspects of first- and second-language teaching is based on parsing technology [17]. A *parser* computes the syntactic structure of input sentences, possibly in combination with their semantic content (provided that all words in the sentence are in the vocabulary, that the grammar available to the system covers all constructions mastered by the student, and that the input does not contain any errors). However, these systems struggle with ungrammatical input and need special measures preventing the parsing quality from getting unacceptably poor. For example, in the *FreeText* system [22], the syntactic–semantic analysis is supplemented with *constraint relaxation* and *sentence comparison.* Other systems invoke matches with corpus texts [6]. Yet another option is the addition of malrules to cover frequent errors [8].

In contrast to a parser, a *generator* produces a sentence or a set of paraphrases from an abstract representation of the content, often called *logical form* (see [27], for an authoritative overview of sentence and text generation technology). In the case of *paraphrase generation*, the generator delivers all possible ways of linguistically realizing the input logical form, given the lexicon and the grammar rules. Virtually all recent natural language generation systems work in a *best-first* manner, i.e., produce only one output sentence rather than the set of all paraphrases. As it is not easy to change the control structure of such a system, the choice of generators is very limited. The paraphrase generator deployed in COMPASS-II [15] does not take logical forms as input but a set of "lexical treelets" as defined in the Performance Grammar formalism [19], which are connected via dependency links. It delivers all possible sentences licensed by the grammar (see Section 4). Probably the first *generator*-based software tool capable of evaluating the grammatical quality of student output was developed by Zamorano Mansilla [30], who applied a sentence generator (KPML [2]) to the recognition and diagnosis of writing errors (“fill-in-the-blank” exercises). Zock & Quint [31] converted an electronic dictionary into a drill tutor. Exercises were produced by a goal-driven, template-based sentence generator, with Japanese as the target language. More recently, Harbusch, Itsova, Koch & Kühner [9], [10] developed the “Sentence Fairy”—an interactive tutoring system for German-speaking elementary schoolers aged 8-10 years which supports writing little stories in L1. The pupils perform limited tasks such as combining simple clauses into compound or complex sentences [25]. A sentence generator calculates all correct paraphrases, and an avatar (the Sentence Fairy; see next section) provides feedback. (This generator is the first version of COMPASS described in [15]; see also Section 4 for the recent version called COMPASS-II, which allows learners of German to generate their own sentences supported by immediate feedback to every step [16], [14].)

Both the Sentence Fairy and the COMPASS-II system presuppose a minimum level of explicit grammatical knowledge in the student. Without it, the feedback information provided by the systems would be incomprehensible. Hence, systems of this type—but also parsing-based systems that are able to elucidate the parse trees they deliver—can only be used in the context of courses where the necessary grammatical concepts, structures and rules have been, or are being, explained. Although this requirement entails a restriction on the range of potential users, in view of the increasing *grammatical awareness* in present-day language instruction (cf. [21], [28]), we believe this drawback is a minor one.

3 THE SENTENCE FAIRY SYSTEM: A VIRTUAL WRITING CONFERENCE SYSTEM FOR ELEMENTARY SCHOOLERS

The Sentence Fairy system (for more details, see [9], [10]) is designed for use by elementary school children aged 8-10 years. Accordingly, we chose yellow/orange/green as “young” colors and made the dialogue as simple as possible. The overall screen layout is the same for all exercises, so that the students’ expectations after an initial learning phase are met and working with the system becomes easier and faster in the long run. In Figure 1, the meta-concept of the screen layout is illustrated by blue balloons. All concepts are distributed over the page according to their procedural order and follow a natural reading direction for German, i.e. left to right, top down, Accordingly, the task to be performed by the pupil is shown in the top left panel (headline of the screen) followed by the exercise itself – distributed over virtually the whole screen.

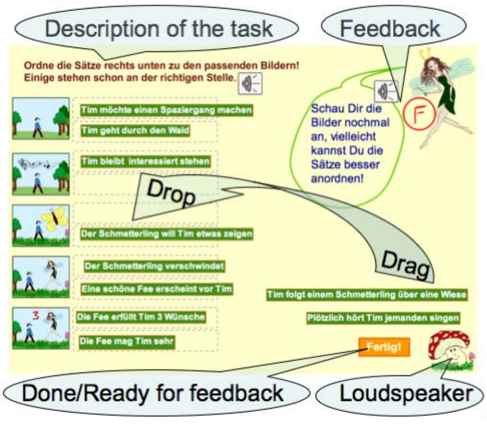

Figure 1. Screen layout of all exercises (cf. blue balloons; German text can be ignored here).

The system’s dialog acts (feedback and audio support) are produced by agents/tutors occupying the corners in the right panel. This layout suggests that the Sentence Fairy and her assisting mushroom only want to help if asked. The Fairy as personalization of the tutor [26] permanently resides in the upper right corner. The Fairy provides feedback when the pupil presses the orange *Fertig* ‘Done’ button in the lower right panel at the end of every exercise. As for audio support, a loudspeaker represented by the mushroom in the lower right corner assists the Sentence Fairy. Sentences of the story printed on the page can be dragged to the mushroom to be read aloud to the pupil. Moreover, exercise texts and feedback are associated with audio-buttons. This helps children with reading problems. The center of the screen is dedicated to the individual drag & drop tasks, in fulfillment of the design constraint that the children should not need to type.

The Sentence Fairy serves as a virtual writing conference, i.e. it plays the role of a communication partner during the process of improving a given text. Taking into account the age of our users, we rely on the concepts of drill-and-practice and tutoring as pedagogical concepts insofar as control is with the system rather than the user. Accordingly, the Sentence Fairy sets up exercises and presents them to the pupils in a fixed order. Within the individual exercises, we aim at authentic problem solving in line with what happens in a real writing conference in the classroom [7]. In our system, the Sentence Fairy (like the elementary school teacher) tutors the session. We model this tutoring dialog in terms of specific *feedback* types [23]. In the three individual exercise types (see below), we work with different feedback types depending on the difficulty level of the exercise type.

The current prototype of the Sentence Fairy system comprises three exercise types: (A) Story reconstruction, (B) Sentence combining, and (C) Word ordering. Exercise type A realizes the first step in a writing conference, i.e. to read the text aloud and clarify the story. Types B and C belong to the text revision process, i.e. the stylistic reshaping of all sentences of the story. By design, spelling checking is not necessary in our drag & drop-based system. The two final steps in a writing conference are not covered in our current prototype, namely final editing, which is supposed to be carried out by the teacher, and pretty printing, which takes the form of hand-writing exercises by the individual pupils.

Type-A Exercises: Story reconstruction. The system randomly extracts a subset of sentences from the story, replacing each member of this subset with empty boxes. The resulting incomplete story is presented on the left side of the screen as a cloze test; the extracted sentences are shown on the right side of the screen and can be dragged into the empty boxes. The automatic feedback is binary here (OK vs. encouraging to do it again, i.e. type-3 feedback according to Mason & Bruning [23]: answer yes/no until correct) and involves matching the pupil's choice with the temporal order in the system's database. In case of negative feedback, the Sentence Fairy gives no concrete hints, because the task is supposed to be easy. So, we decided to keep the pupil motivated to go on without specific hints.

Type-B Exercises: Sentence combining. The system shows alternative Coordinating and Subordinating Conjunctions for two consecutive clauses in a container. The clauses are presented in various word orders in the right lower panel. In order to practice building a compound sentence, the student selects a Conjunction and two clauses with appropriate word orders, and moves them into the corresponding choice boxes. Based on the list of automatically generated paraphrases (provided by COMPASS-II; cf. Section 4), the system computes its feedback: "OK" if the student response matches one of the system's paraphrases, or some help otherwise. In the latter case, a feedback text is presented, indicating which syntactic constraint(s) imposed by the chosen Conjunction was/were violated, e.g. "You used Main Clause word order with a Subordinating Conjunction", which represents so-called type-7 feedback according to [23]: explanation of a malrule. The system is also able to evaluate the application of pronominalization rules, e.g. to replace one of two coreferential NPs with a personal pronoun.

Type-C Exercises: Word ordering. The system randomly selects a simple or compound clause and presents it on the screen after having replaced the major phrases of one of the original clauses with empty boxes. In the first example, a Relative Clause has to be produced by the pupils. The system has automatically predetermined*: Tim, der _ _ _ _, bleibt interessiert stehen* 'Tim, who _ _ _ _, stands still interested'). The pupil has to find an appropriate word order for the Relative Clause (given the words: 'somebody', 'sing', 'suddenly', 'hears'). The phrases are presented on the screen as selectable items, and the student is invited to assemble a correct sentence—possibly one with a constituent order that differs from the original order—by dragging the phrases into the empty boxes. The Sentence Fairy can evaluate grammatical correctness by matching the resulting sentence against the set of paraphrases computed by the COMPASS-II system. Accordingly, an appropriate feedback is selected from a fixed list of error messages or compliments. Our generator is able to evaluate the correctness of subtle German word order variants in varying syntactic contexts (*singen hört* 'sing hears' vs. *hört singen* 'hears sing' 'hears sing').

4 THE SYNTACTIC STRUCTURES UNDERLYING THE PARAPHRASES GENERATED BY THE COMPASS-II SYSTEM

COMPASS-II (for more details see [16], [14]) implements a paraphrase generator for the linguistic formalism of *Performance Grammar* [19]. In the new user interface, learners of German as L2 can write their own sentences in a drag-and-drop manner using visualized syntactic structures and get accurate feedback at any stage (*scaffolded writing;* [16], [14]). In this paper, we focus on the interface with the teacher mode, i.e. the *internal* representation of sentences in COMPASS-II, which allows producing all paraphrases. In Section 5, we outline how this information will be specified in the teacher mode *without* any knowledge of the underlying Performance Grammar formalism.

The *Performance Grammar (PG)* formalism distinguishes three aspects of the structure of sentences: *dependency* relations (grammatical functions), *constituent* structure, and *linear* order. The dependency relations and the constituent structure together form the *hierarchical* (or *dominance*) structure. The dependency relations include functional relations, i.e. grammatical functions (Subject, Direct and Indirect Object, Head, Complement, Determiner, Modifier, etc.). The constituent structure comprises word categories (parts of speech) and word groups (the various types of phrases and clauses). PG's hierarchical structures can be visualized as rather flat unordered trees, as depicted in Figure 2. The application of linear order rules may give rise to structures that can be visualized as ordered trees with crossing branches (graphs; see Figure 4 for the linearization of a complete sentence). The split between hierarchical/dominance and linear structure allows entering unordered structures as input (e.g., underspecified with respect to sentence type, e.g. Interrogative vs. Declarative Main Clause). Based on such unordered trees, COMPASS-II produces all word order variants as paraphrases (for reasons of space, we skip details of how word ordering in PG works; but see [11]).

Figure 2 depicts an *elementary treelet* (also called *lexical frame)* for the word form *Junge* 'boy'. The rightmost branch specifies the lexical anchor of the treelet: *Junge* is a n[oun] functioning as the h[ea]d of a N[oun]P[hrase]. The second layer of nodes represents *grammatical functions*: det[erminer], q[uantifier], mod[ifier], etc. The third layer consists of phrasal nodes that specify which types of constituents are allowed to fulfill the function above them (the slash '/' separates alternative options). For example, the modifier role can be played by a P[repositional]P[hrase], an A[djectival]P[hrase], or a S[entence] (more precisely, a Relative Clause). One node in the third layer specifies the word category of the h[ea]d, i.e. the lexical anchor (here n[oun]). Every node of a lexical treelet has associated with it a set of *morphosyntactic features*. They are specified in the lexicon of word forms[1]. A *feature* is a combination of a property and a *value* specification (*Typed Feature Unification*—widely used in theoretical and computational linguistics, e.g. in HPSG [29]). The latter may be a single term (which holds for the features of the Noun *Junge*, with the feature-value pairs: wordform=Junge, lemma=Junge, gender=masculine, person=3rd, case=nominative, and number=singular as depicted in Figure 2), but it may also be a disjunctive set of alternative value options. For instance, the word form *Jungen* (for 'boy' or 'boys') has the same treelet associated with it, except for the leaf node *Jungen*. However, the feature structure for the noun *Jungen* expresses the fact that *Jungen* can have *Genitive* or *Dative* or *Accusative case* if and only if its *Number* is *Singular,* whereas it can have *Nominative, Genitive, Dative* or *Accusative case* if and only if its *Number* is *Plural.* In disjunctive feature structures, the alternative value options are enumerated within curly brackets (the logical inclusive OR), and square brackets enclose an AND enumeration. The feature specification for the word form *Jungen* at node n[oun] now looks as follows:

> [wordform=Jungen AND lemma=Junge AND gender=masculine AND person=3rd AND{[case={gen OR dat OR acc} AND number=singular] OR[case={nom OR gen OR dat OR acc} AND number=plural]}]

Phrasal leaf nodes (*foot nodes*) can be expanded by an appropriate treelet whose root node carries the same label, thus forming more complex phrases. This operation (technically called *Unification*) merges a foot node of one treelet with the root node of another treelet (see Figure 3).

---

[1] *Word forms* are members of an inflectional paradigm. For instance, *Junge* and *Jungen* both belong to the same paradigm: the paradigm of the "lemma" *Junge*. Lemmas are referred to by one member of the paradigm—e.g., *Junge* (Nominative, Singular).

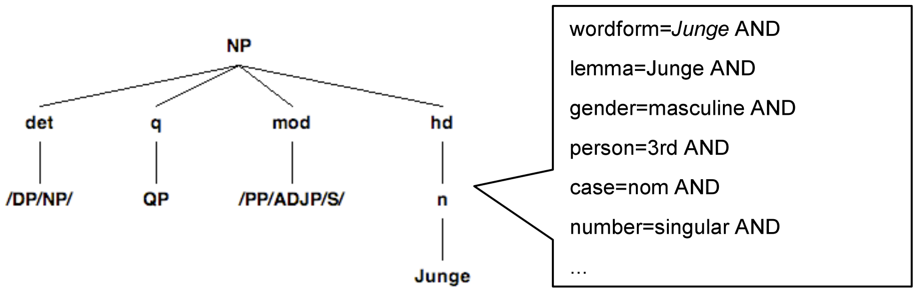

Figure 2. Lexical frame illustrating the grammatical function specification on layer two and depicting features at nodes for selected node *n*.

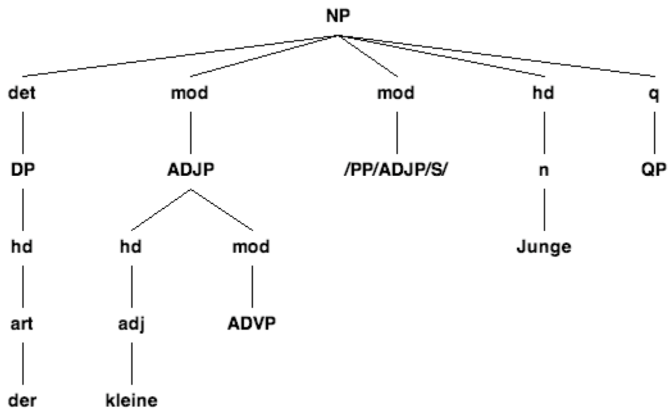

Figure 3. Expansion of the DP and ADJP nodes of the lexical frame of *Junge* 'boy'. The q[uantifier] and mod[ifier] functions remain empty (phrasal foot). COMPASS-II ignores such (facultative) branches.

In Figure 3, the D[eterminer]P[hrase] foot node has been expanded by the DP root node dominating the appropriate Masculine Definite Article *der*, and the ADJ[ective]P[hrase] root node dominating the word form *kleine* 'small' expands the foot node of a mod[ifier] branch. Whether a root and a foot node can be merged ("unified") or not, depends not only on their label but also on the associated features. The feature specifications are used by the unification operation to select legal expansions. For instance, the fact that S-type modifiers within NPs should be Relative Clauses (rather than, say, Main Clauses) is controlled by features. Similarly, other features control the selection of the inflected word form *kleine* 'small' instead of the uninflected *klein*. For details of the unification process, in particular on how it deals with phenomena of grammatical agreement, we refer to the papers quoted above.

For dominance trees, COMPASS-II can produce all word order variants based on word-order rules (see [11] for PG's word ordering in Dutch, English and German). The root of every treelet is associated with a *topological array*. Constraints specify which constituents are allowed in a slot (N.B. there are slots hosting several constituents; cf. slot M6). In German, the S node is associated with nine array slots named F1, M1, M2, M3, M4, M5, M6, E1, E2. The slot names are inspired by the theory of *Topologische Felder* in German structural linguistics, i.e. F = Forefield, M = Midfield, E = Endfield—with the following slot constraints (Figure 4 depicts one option for the interrogative pharaphase *Was will der kleine Junge dass ich sage* 'what does the little boy want me to say'):

F1: Subject, topic or focus in a Declarative Main Clause (one constituent only); a wh-constituent (a phrase including an Interrogative Pronoun) in an Interrogative Main Clause; a wh-constituent in a Complement Clause;

M1: Finite Verb in a Main Clause; the Complementizer *dass* 'that' of a Complement Clause;

M2-M5: Non-wh Subject, Direct Object, Indirect Object, Non-finite Complement Clause, Modifier;

M6: Finite Verb, possibly preceded by Particle and Pre-infinitival *zu* 'to', in a Subordinate Clause;

E1-E2: Nonfinite Complement preceding Finite Complement.

Peripheral slots of neighboring topologies can be *shared* (cf. F1:*was* in Figure 5; details skipped here).

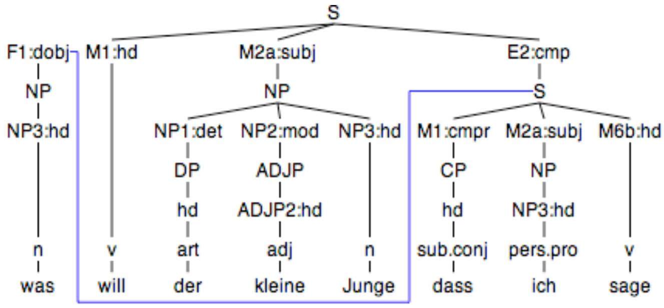

Figure 4. Graphical representation of the word order process filling topological arrays (by *sharing* of peripheral slots like F1, *was* is fronted). Unfilled slots are skipped in the graph.

5 THE TEACHER MODE TO SET UP THE INPUT FOR THE PARAPHRASE GENERATOR COMPASS-II

Through the *teacher mode*, users with elementary linguistic knowledge of the constituent structure of simple clauses (but unfamiliar with any linguistic grammar formalism) can enter their own stories. COMPASS-II will translate these into e-learning exercises fully automatically via the generation of paraphrases to be selected as exercises.

The teacher mode consists of three stages: (1) construction of dominance trees for simple sentences (cf. Section 5.1), (2) specification of rhetorical relations (cf. Section 5.2), and (3) selection of lexical choice variants from a suggestion list generated by WORDNET (cf. Section 5.3).

5.1 Dominance-tree Construction

The dominance trees for COMPASS-II might be constructed by parsing all sentences of the story entered by the teacher. However, hardly any sentence can be parsed unambiguously (cf. the proverbial *Time flies like an arrow,* for which Wikipedia lists no less than seven different interpretations). Hence, it is notoriously difficult to produce highly reliable feedback based on parsing results (cf. Section 2). In order to eliminate syntactic ambiguity as much as possible, we decided on a dialogue between user and system where constituents classified according to their basic grammatical functions such as Subject and (In)direct Object—which teachers are supposed to know anyway—are entered by the teacher. In order to keep the length of such dialogues within reasonable limits, we do use a syntactic parser to process subclausal constituents in the form of Noun Phrases (NPs) or Prepositional Phrases (PPs). Parsing such phrases tends to be unproblematic (see the example in Figure 5, with a complex Subject *der kleine Igel* 'the little hedghog'). Pushing the OK-button to the right of the input field starts building a Performance Grammar-based dominance tree in the teacher mode component (cf. lower panel of Figure 5, which remains hidden to the user). After the user pushes the OK-button, all word forms of the phrase the teacher has entered are searched for in the lexicon (CELEX [1]). In NPs and PPs, the system determines the Head unambiguously (e.g. for the Noun Phrase *der kleine Igel* 'the little hedghog', *Igel* is identified as the Head Noun due to its position even if a Relative Clause follows). The lexical frames for all other entered words of the phrase get unified according to the phrasal nodes occurring in the lexical frame of the Head (e.g., *der* 'the' goes to D[eterminer] P[hrase]; *der* in its Relative Pronoun meaning can be ruled out by word order inspection). In the example depicted in Figure 5, the Determiner and the Adjective Phrase become attached to the dominance structure. During this construction process, the whole dominance tree is tested for correctness of all morphosyntactic features by procedures that are also used in COMPASS-II (e.g., *der kleiner Igel* would cause an error, and the teacher would be asked to revise the input—which is supposed to be a typo by a native speaker of German). Associated with every grammatical term is a *mouse-over function* that provides the definition of the term, thus preventing confusion between terms with similar meanings, such as Accusative Object vs. Direct Object.

For many word forms, the lexicon specifies more than one lexical frame (e.g., *bauen* 'build/construct' can be intransitive, transitive or ditransitive, with differing sets of obligatory grammatical functions). In such cases, a list of alternative lexical frames is presented from which the user can select the intended option. The system then presents the lexical-frame branches that correspond to the grammatical functions licensed by the selected frame (Figure 6). For instance, if the transitive version of a verb is selected, the system asks to enter a constituent into the Direct Object position. Additionally, a Modifier of type ADVP (Adverb) can be added. (Other options remain hidden in the choice indicated by left and right arrow (cf. mod[ifier] ADV[erbial]P[hrase] provided as potential next grammatical function to be specified by the teacher) in order not to overtax the teacher's attention while entering the constituent s/he just has in mind.)

Actually, we expect the teacher to enter a series of simple main clauses (no fixed upper or lower limits determined; however, step (2) is only meaningful with at least two clauses), and to describe the rhetorical relations in the next phase. However, the system automatically provides all grammatical functions licensed by a given treelet (lexical frame). For example, the set of constituents of a Subordinate Clause fulfilling the grammatical function of CMP (Complement) includes a Complementizer node to be filled in by the user as described above. The system can also infer abstract RST relations from entered Subordinating Conjunctions in a teacher-specified complex clause instead of choosing the option of two simple clauses in step (1) in combination with a rhetorical relation selected in step (2) (see next section).

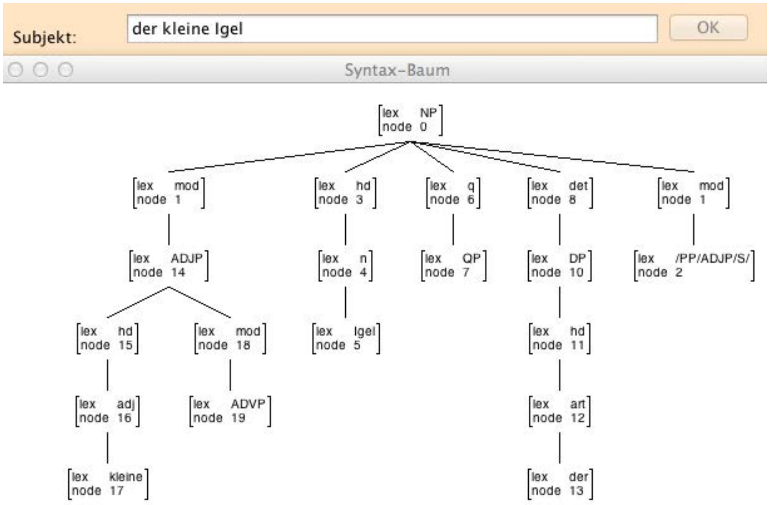

Figure 5. Snapshot of the dominance tree (**NOT** to be inspected by the teacher) for a complete Noun Phrase *der kleine Igel* 'the little hedghog'. This phrase is entered as Subject into the upper panel, which represents a dialogue unit in step (1) of the teacher mode.

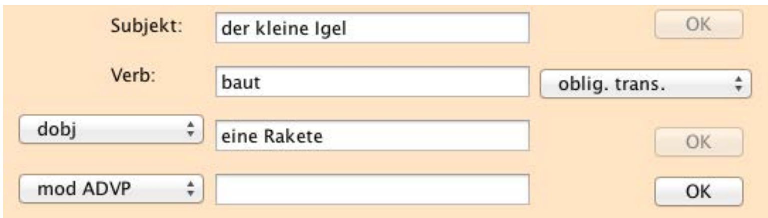

Figure 6. Selection of the transitive variant of the Verb *baut* 'builds' (3rd Person, Singular, Present Tense) leads to the presentation of text entry boxes for the obligatory grammatical function d[irect] obj[ect] (here filled with *eine Rakete* 'a rocket') and a facultative mod[ifier] ADVP.

5.2 Specification of Rhetorical Relations

The information on *rhetorical relations*, currently only *temporal* or *causal relations* are displayed, is asked in a non-technical manner in a second step of the teacher mode. These RST-relations become the basis for a variety of sentence combination exercises (cf. [25]). Alternatively to marking a box in dialogue step (2), a complex clause including a Subordinate Conjunction can be typed in step (1) if preferred to determine the rhetorical relation. Irrespective of the way RST-relations are entered, the system translates the choice into an abstract relation in the *Rhetorical Structure Theory (RST* [24]). For instance, "Peter combed his hair **before** Peter left the house" will become EARLIER(a,b), which can also result in "Peter left the house **after** he combed his hair ".

Figure 7 illustrates the non-technical dialogue for the two sentences *Anja baut eine Rakete* ‘Anja builds a rocket’ and *Anja will auf den Mond fliegen* ‘Anja wants to fly to the moon’. The teacher can select whether sentence 1 is earlier than *(vorzeitig*) and/or the reason (*Grund*) for sentence 2. In our example the teacher has chosen a CAUSALITY relation, which elicits sentence combinations using Subordinating Conjunctions such as *weil* and *da* ‘because’ in various word orders.

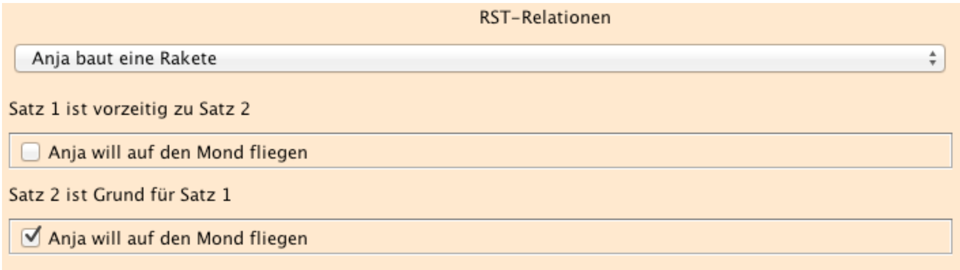

Figure 7. Illustration of dialogue box to determine RST-relations without using technical terms.

5.3 Lexical choice variation

In the third step of the teacher mode, *lexical choice* alternatives for all words in a sentence can be added (e.g. *run, hasten, hustle*). To this purpose, the system automatically generates, from WORDNET [5], a *suggestion list* consisting of synonyms, hyponyms and hyperonyms. As this requires translating the German terms to the English WORDNET and translating the retrieved words back again, the teacher can erase wrong translations for unintended meanings of the German term, or add alternatives not provided in the suggestion list.

We have skipped many details of the dialogues of the teacher mode here for reasons of space. At the end of step (3), the system writes to a file an SGML-formatted dominance structure, with RST-relations and alternative lexical items at the word level. This output file can be read into COMPASS-II, where all possible sentence combinations and all possible word-order variations are calculated. From this list, complex sentences for Sentence Fairy exercises can be selected.

6 DISCUSSION

We view the current version of the Sentence Fairy as the prototype of an “engine” that can drive the automatic evaluation and diagnosis of sentences produced by L1 students of German. The system is far from complete and not yet usable in the classroom. Several software aspects are in need of improvement, in particular the robustness of the system. We hope, however, that the foregoing description rouses the interest of the (I)CALL community in the great potential of generator-based systems as providers of online L1 and L2 writing support to students.

We are keenly aware that our exact feedback makes heavier demands on the student’s explicit grammatical knowledge than many other writing-support tools. However, in quite a few languages, rules for spelling and other aspects of writing presuppose that the writer is able to explicitly recognize detailed syntactic properties of the sentence under construction (which are taught already in elementary school). Well-known examples are morphosyntactic distinctions that got lost in pronunciation but are maintained in spelling—e.g., the distinction in German between *dass* (Subordinating Conjunction) and *das* (Determiner or Pronoun), We suggest that NLP-based tools can be employed fruitfully in integrated writing and grammar courses.

A particularly useful approach to teaching grammar and writing in an integrated fashion—one that is relatively easy to implement in the Sentence Fairy and COMPASS-II—is to focus on an interrelated set of syntactic constructions and the rules controlling their shape. An example concerns coordinate structures and their elliptical forms (which elementary schoolers would rarely produce without stimulation): *forward conjunction reduction, gapping, right node raising*, etc. Recently, we have laid the PG-oriented linguistic and computational groundwork for these constructions, which have very high usage frequencies ([18], [11], [13]). One of the topics we might address in the near future is to integrate such exercises into the Sentence Fairy.

ACKNOWLEDGEMENT

We are indebted to Gergana Itsova and Christine Kühner for their collaboration in developing the Sentence Fairy; and to Gerard Kempen and Theo Vosse for helping in the development of COMPASS-II. Moroever, we thank Gerard Kempen for his valuable comments on previous versions of this paper.

REFERENCES

[1] Baayen, R.H., Piepenbrock, R. and Gulikers, L. (1995). The CELEX lexical database (release 2.5, CD-ROM). Linguistic Data Consortium: University of Pennsylvania, PA.

[2] Bateman, J.A. (1997). Enabling technology for multilingual natural language generation. Natural Language Engineering, 3, pp. 5–55.

[3] Davies, G. and Bruzzone, M. (2004). Fun with Texts Version 4.0 (new multi-media version), Maidenhead: Camsoft [See http://www.camsoftpartners.co.uk/fwt.htm.]

[4] Delmonte, R., Delcloque, P. and Tonelli, S. (Eds.) (2004). Proceedings. of the InSTIL/ICALL2004 Symposium, Venice (Italy). [See http://www.isca-speech.org/ archive/].

[5] Fellbaum, C. (Ed.) (1998). WordNet – An Electronic Lexical Database. Cambridge/MA: The MIT Press.

[6] Granger, S. (2004). Computer learner corpus research: Current status and future prospects. In Connor, U., & Upton, T. (Eds.). Applied Corpus Linguistics: A Multidimensional Perspective (pp. 123–145). Amsterdam: Rodopi.

[7] Graves, D.H. (1983). Writing: Teachers & Children at Work. Portsmouth, NH: Heinemann.

[8] Fortmann, C. and Forst, M. (2004). An LFG Grammar Checker for CALL. In [4] , pp. 59–61.

[9] Harbusch, K., Itsova, G., Koch, U. and Kühner, C. (2008). The Sentence Fairy: A natural-language generation system to support children's essay writing. CALL Journal, 21:339–352.

[10] Harbusch, K., Itsova, G., Koch, U. and Kühner, C. (2009). Computing accurate grammatical feedback in a virtual writing conference for German-speaking elementary-school children: An approach based on natural language generation. CALICO Journal, 20, pp. 626–643.

[11] Harbusch, K. and Kempen, G. (2002). A quantitative model of word order and movement in English, Dutch and German complement constructions. In Tseng, S.-C. (Ed.), Proceedings of the 19th COLING, Taipei, ROC (pp. 328–334). San Francisco, CA: Morgan Kaufmann.

[12] Harbusch, K. and Kempen, G. (2006). ELLEIPO: A module that computes coordinative ellipsis for language generators that don't. In Proceedings of the 11th EACL, Trento, Italy, (pp. 115–118). East Stroudsburg, PA: ACL. [See http://www.aclweb.org/anthology/E/E06/#2000].

[13] Harbusch, K. and Kempen, G. (2007). Clausal coordinate ellipsis in German. In Nivre, J., Kaalep, H.-J., Muischnek, K. and Koit, M. (Eds.), Proceedings of the 16th NODALIDA, Tartu, Estonia (pp. 81–88). [See http://dspace.utlib.ee/dspace/handle/10062/2683].

[14] Harbusch, K. and Kempen, G. (2011). Automatic online writing support for L2 learners of German through output monitoring by a natural-language paraphrase generator. In: Levy, M. Blin, F., Bradin Siskin, C. and Osamu, T. (eds.), WORLDCALL - International Perspective on Computer-Assisted Language Learning,Routledge, New York, NY, USA, 2011, pp. 128-143.

[15] Harbusch, K., Kempen, G., van Breugel, C. and Koch, U. (2006). A generation-oriented workbench for Performance Grammar. In Colineau, N., Paris, C., Wan, P., & Dale, R. (Eds.), Proceedings of the 4th INGL, Sydney, Australia (pp. 9–11). Retrieved from http://www.aclweb.org/anthology/W/W06/W06-14.pdf.

[16] Harbusch, K., Kempen, G. and Vosse, T. (2008). A natural-language paraphrase generator for on-line monitoring and commenting incremental sentence construction by L2 learners of German. In Koyama, T., Noguchi, J., Yoshinari, Y. and Iwasaki, A. (Eds.), Proceedings of WORLDCALL, Fukuoka, Japan (pp. 190–193).

[17] Heift, T. and Schulze, M. (Eds.) (2003). Error diagnosis and error correction in CALL. CALICO Journal, 20(3).

[18] Kempen, G. (2009). Clausal coordination and coordinate ellipsis in a model of the speaker. Linguistics, 47, 653–696.

[19] Kempen, G. and Harbusch, K. (2002). Performance Grammar: A declarative definition. In Theune, M., Nijholt, A. and Hondorp, H. (Eds.), Computational Linguistics in the Netherlands 2001 (pp. 146–162). Amsterdam: Rodopi.

[20] Kempen, G. and Harbusch, K. (2003). Dutch and German verb constructions in Performance Grammar. In Seuren, P.A.M., & Kempen, G. (Eds.), Verb Constructions in German and Dutch (pp. 185–222). Amsterdam: Benjamins.

[21] Levy, M. (1997). CALL: context and conceptualization. Oxford: Oxford University Press.

[22] L'haire, S. and Vandeventer Faltin, M. (2003). Error diagnosis in the FreeText project. CALICO Journal, 20(3), 481–496.

[23] Mason, J.B. and Bruning, R. (2001). Providing feedback in computer-based instruction: What the research tells us. [See http://dwb.unl.edu/Edit/MB/MasonBruning.html].

[24] Mann, W.C. and Thompson, S.A. (1988). Rhetorical Structure Theory: Toward a functional theory of text organization. Text, 8:243-281.

[25] Mellon, J.C. (1969). Transformational sentence-combining: A method for enhancing the development of syntactic fluency in English composition. Urbana, IL: National Council of Teachers of English.

[26] Paechter, M. and Schweitzer, K. (2006). Learning and motivation with virtual tutors. Does it matter if the tutor is visible on the Net? In: Pivec, M. (Ed.), Affective and emotional aspects in Human-Computer Interaction. Amsterdam: IOS Press.

[27] Reiter, E. and Dale, R. (2000). Building applied natural language generation systems. New York: Cambridge University Press.

[28] Roehr, K. (2007). Metalinguistic knowledge and language ability in university-level L2 learners. Applied Linguistics, 29, pp. 173–199.

[29] Sag, I.A, Wasow, T. and Bender, E. (2003). Syntactic Theory: A Formal Introduction (Second edition). Stanford, CA: CSLI Publications.

[30] Zamorano Mansilla, J.R. (2004). Text generators, error analysis and feedback. In [4], pp. 87–90.

[31] Zock, M. and Quint, J. (2004). Converting an Electronic Dictionary into a Drill Tutor. In [4], pp. 20-23.