GI-TUTOR: Grammar-Checking for (Italian) Students of German

Rodolfo Delmonte, Alessia Battisti

Department of Linguistic Studies & Department of Computer Science
Ca' Bembo, Dorsoduro 1075
Ca' Foscari University - 30123 Venezia (Italy)
website: project.cgm.unive.it
delmont@unive.it, alessia4battisti@gmail.com

Abstract

Computers have played a significant role in second language learning as they can offer different types of language activities and provide learners with immediate and appropriate feedbacks. In ICALL systems, learning is activated by focusing the learner's attention to the correct form and comparing it to the wrong one. Feedback offers an explicit explanation of the mistakes made by the student. The focus of this paper is a grammar checker designed for Italian native speakers learning German - GI-tutor. The system's structure was taken from a previous study [28] and enforced to analyze a conspicuous number of sentences, with different sentence and phrase structures. The lexicon used has been manually organized at the beginning, with some 8,000 entries overall; then, an enlargement has been obtained through an adaptation of the lexicon made available by Hamburg University Constraint Dependency Grammar (JWCDG) and downloaded from their website1. A corpus containing wrong sentences was expanded by extracting data from exams written by first-year students of the German course at the University Ca' Foscari. The errors were then classified in order to obtain a general statistical analysis of the main problems encountered when learning German.

Attention was given also to parsers and their use and functionality for language learning. Furthermore, the performance of the constituency grammar checker was evaluated to determine the types and frequencies of errors it can successfully diagnose. This was done by comparing it to ParZu - a generic German dependency parser developed at the University of Zürich2 and to Stanford Parser for German.

1. Introduction

We present GI-Tutor (GermanItalian-Tutor) a grammar checker and learning environment for Italian – and other Romance languages - students of German (see [3;4;5;19]) which is based on the shallow parser of Italian used to produce the syntactic constituency for the Italian Treebank called VIT (Venice Italian Treebank, see [6;8]). The output of the parser is a bracketing of the input tagged word sequence, which is then passed to the higher grammatical functional processor. This is a Lexical Functional Grammar (LFG)-based (see [2]) c-structure to f-structure mapping algorithm which has three tasks: the first task is to compute features from heads; the second is to compute agreement, and the third is to impose LFG's grammaticality principles of coherence and consistency to insure that the number and type of arguments are constrained by the lexical form of the governing predicate.

The parser uses a recursive transition network (RTN) which has been endowed with a grammar and a lexicon of German of about 8,000 entries. The lexicon has then been enlarged by an adapted version of JWCDG’s Lexicon, about 700,000 entries. The grammar is written in the usual arc-transition nodes formalism, well known in augmented transition networks (ATN). However, the aim of the RTN is to produce a structured output both for well-formed and ill-formed grammatical sentences of German. To this end, we allowed the grammar to keep part of the rules of Italian at the appropriate structural level. For example, Italian speakers are inclined not to use V2 structures, which are checked at sentence level. Grammar checking is accomplished at different levels of constituency structure, and also at the function-structure level.

2. The Cascaded Shallow Parser

The function of the shallow cascaded parser is to create syntactic structures eligible for grammatical function assignment. This task is made simpler given the fact that the disambiguator associates a net or constituency label with each disambiguated tag. Parsing can then be defined as a bottom-up collection of constituents which contain either the same label or which are contained in or are a member of the same net or higher constituent. No attachment is performed in order to avoid being committed to structural decisions which might then reveal themselves to be wrong. We prefer to perform some readjustment operations after structures have been built rather than introducing errors from the start. Readjustment operations are in line with the LFG theoretical framework which assumes that f-structures may be recursively constituted by subsidiary f-structures (i.e., by complements or adjuncts of a governing predicate). Therefore, the basic task of the shallow parser is that of building shallow structures for each safely recognizable constituent and then pass this information to the following modules.

The tagset we use for German consists of 85 tags which encode a number of important features for the parser that we need to produce syntactic and semantic feedback information. These features include transitivity, modality, and auxiliary class for verbs as well as semantic classes like colour, human, and evaluative for nouns. Tags are disambiguated by a statistical and syntactic procedure which is set up for special ambiguity classes. In some cases, we use appropriately organized Finite State Automata, which capture fixed structural patterns of German grammar. The output of the disambiguator is a partially disambiguated input which is
then processed by the shallow cascaded parser. Here below is an example of a wrong sentence:

```
ex1. mein opa bringt ihr sie.
```

The parser issues two error messages. The first one regards case assignment: "sie" is in the accusative whereas dative is required. The second one concerns the position: "ihr" is the dative clitic and should come after the accusative clitic. In order to recognize errors, full morphological and lexical subcategorization information for all words must be available.

### 2.3 GI-Tutor and Tutoring Systems

GI-Tutor addresses issues for students of German who are enrolled in degree programs in Language Sciences where General Linguistics and other similar courses are required. The system is used mainly to check the correctness of assignments for students of German where they are required to produce a composition, i.e. to describe freely some specific and exceptional situation or detail of their daily life. Then, students who want to improve their knowledge of German are directed to guided exercises.

A standard spelling and grammar checker shares with an ILTS system the focus on identifying errors, but they are based on presuppositions about typical errors made by native speakers which don't refer to language learners (see [30]). Furthermore, Rimrott and Heift (see [34:73]) notice that "in contrast to most misspellings by native writers, many L2 misspellings are multiple-edit errors and are thus not corrected by a spell checker designed for native writers". A comparison between parser-based CALL and 'conventional CALL' (see [16]) shows that in parser-based CALL, students have a relative freedom of writing activity and can thus compose a potentially large amount of sentences. With the use of ICALL tools, production skills are practiced much more, as word and structures recalling and constructing activities are required. The limitation of the parser is, however, its focus on the syntax of textual input. Another limitation could be its characteristic "not to be foolproof"; the parser can not analyse all the syntax of a language in an accurate way (see [19]), especially if dialect usages are in the student structures.

In order to build exercises automatically, we duplicated all the sentences with mistakes from our database and created the corresponding correct sentences. This procedure allowed us to generate exercises for students by picking at random a certain number of sentences, say three or four, from the correct subset and mix them with one or two sentences from the mistakes subset. The task for students could be either to identify the sentences with error(s) or correct the errors). In either case, their response could be easily checked. Rather than discussing these exercises, we will concentrate on the "Sentence Creation" exercise which requires students to produce a correct sentence from a sequence of input hints consisting of lemmata (uninflected content words). This procedure starts by selecting randomly one of the correct sentences. It then deletes the function words in the sentence
and displays the lemma for each content word. The resulting sequence of words is presented to students who are asked to build a correct sentence. Given the fact that students can produce any sentence using the lemmata provided, we cannot evaluate their response by a simple pattern-matching operation. The system has to check for correctness.

The tutoring module of the system requires the student to create his/her profile with credentials which are then used by the system to recover previous results that have been recorded in a specific directory dedicated to the student and the teacher. We use Italian for student instructions to allow Italian students to use the system. We also prompt students not to type upper case letter because the system only uses lowercase letters. Students are asked to repeat an exercise after they have checked for mistakes in the feedback window. In the case of a sentence being correctly entered, the system simply confirms the correctness and proposes a new sentence. Whenever students decide to interrupt the exercise, an evaluation is issued for the whole interaction, and the result is shown graphically by turning previous successes and failures into scores and then transforming scores into coloured bars: red for mistakes and green for correct sentences. A comment is generated based on the severity of the errors and on the basis of the overall score. On any subsequent repeated access by the same student, the system will ask whether it intends to continue from previous work or to start again from beginning.

3. Types of Errors in the Training Corpus

As Wagner et al. [39] affirm, the creation of a corpus of ungrammatical sentences normally requires time and deep linguistic knowledge. In GI-Tutor linguistic knowledge is comprised in a database of ungrammatical and equivalent grammatical sentences that represent the corpus used.

A relative small corpus of errors was collected in a previous study (see [37]), which represented the most frequent errors that Italian native speakers make when studying German. Lately, the corpus has been greatly enhanced by extracting anonymously, ungrammatical sentences from exam papers written by first-year students of German at the University Ca’ Foscari. Then, these sentences have been digitalized. The corpus includes in total 1262 sentences: 631 ungrammatical sentences and 631 corresponding correct sentences. The ungrammatical ones can contain more than one error.

In the corpus we can get a general view of typical errors that Italian native speakers make when studying German. Yet the concept of error is very subjective (see [26]); for example, a German native speaker can consider an utterance as acceptable, even if it is not completely understandable. Collected sentences can be considered unusual because of the syntax and the content. Besides, some errors have to be considered context-sensitive.

Furthermore, it has to be considered that the sentences have been written by students that have a limited vocabulary and a poor linguistic knowledge of German. Despite the limited amount of data, a small statistical analysis of different errors types has been carried out in order to determine the area of most significant deficits of the learners. As it can be observed from Table 1, the highest percentage of errors is concentrated at the syntactic level, as a number of authors have also acknowledged. In his study Juozulynas (see [23]) shows that syntax is the most problematic area, followed by morphology. He points out that 80% student errors are not of semantic origin and thus, are potentially identifiable by a syntactic parser. Our study confirms only partially this hypothesis: 37.85% of collected errors are from the syntactic level and 36.15% is from the morphologic areas which can be partially regarded as semantic in nature.

3.1 Errors classification

In this section, we propose a new form of error categorization, as the treatment in the two theses by [37] and [28] seemed to be rather unclear and too much structured. Errors have been divided into groups, equivalent to the main language levels (see [18;23]); the resultant four groups have been subdivided into subcategories on the basis of certain features. Error categories (see [22;23;25]) contain both ungrammatical sentences, which have been written in brackets, and correct ones. Examples reported below show how they appear in GI-Tutor: clauses are tokenized and the acronyms ‘b1’ ‘c5’ ‘h7’ identify different errors categories and equivalent sentence number. Upper and lower case, umlaut, scharfes S (ß) and punctuation do not assume a significant role in the automatic analysis in GI-Tutor, because in this context they are only orthographic rules and are not important to the parsing strategy. The capital letter was preserved only for proper nouns (forename, surname, geographic name). In this study, we do not observe capitalization errors, and thus they are not taken into account.

<table>
<thead>
<tr>
<th>Error Type</th>
<th>No. Errors</th>
<th>%</th>
<th>No. Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orthographic errors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orthography</td>
<td>36</td>
<td>48.00%</td>
<td>36</td>
</tr>
<tr>
<td>Interference with English</td>
<td>36</td>
<td>48.00%</td>
<td>36</td>
</tr>
<tr>
<td>Interference with Italian</td>
<td>3</td>
<td>4.00%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>11.54%</td>
<td>75</td>
</tr>
<tr>
<td>Lexical errors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error selection of the preposition</td>
<td>58</td>
<td>61.70%</td>
<td>58</td>
</tr>
<tr>
<td>Error selection of the verbal particle ‘zu’</td>
<td>13</td>
<td>13.82%</td>
<td>13</td>
</tr>
<tr>
<td>Wrong choice of auxiliary verb</td>
<td>5</td>
<td>5.32%</td>
<td>5</td>
</tr>
<tr>
<td>Unsuitable semantics</td>
<td>18</td>
<td>19.15%</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>94</td>
<td>14.46%</td>
<td>92</td>
</tr>
<tr>
<td>Morphologic errors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Congruence errors (nouns, pronouns, articles, adjectives)</td>
<td>178</td>
<td>75.74%</td>
<td>169</td>
</tr>
<tr>
<td>Congruence errors (verbs)</td>
<td>43</td>
<td>18.30%</td>
<td>43</td>
</tr>
<tr>
<td>Error inflection of past tenses</td>
<td>14</td>
<td>5.96%</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>235</td>
<td>36.15%</td>
<td>226</td>
</tr>
<tr>
<td>Syntactic errors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Violation of sentence structure</td>
<td>75</td>
<td>30.49%</td>
<td>69</td>
</tr>
<tr>
<td>Violation of V2-rule</td>
<td>69</td>
<td>28.05%</td>
<td>68</td>
</tr>
<tr>
<td>Positional errors of mandatory or optional complements in Mittlefeld</td>
<td>26</td>
<td>10.57%</td>
<td>26</td>
</tr>
<tr>
<td>Positional errors of the negative particle ‘nicht’</td>
<td>38</td>
<td>15.45%</td>
<td>38</td>
</tr>
<tr>
<td>Unsuitable usage of separable and inseparable verbs</td>
<td>33</td>
<td>13.41%</td>
<td>32</td>
</tr>
<tr>
<td>Lack of subject pronoun or subject pro-drop</td>
<td>5</td>
<td>2.03%</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>246</td>
<td>37.85%</td>
<td>238</td>
</tr>
<tr>
<td>Total</td>
<td>650</td>
<td>100.00%</td>
<td>631</td>
</tr>
</tbody>
</table>

Table 1. Typology of errors in the database

3.2 Orthography

A violation of orthographic rules (see [34]) does not reveal any lack of language competence in the broader sense, as the
observation of these rules is related to the written competence. These errors are made mostly because of the interference with another foreign language or the mother tongue.

In this corpus, 36 orthographic errors are attributed to interference with English, since English is usually the first foreign language taught at school and the similarities to German can mislead learners.

```plaintext
fp(b4, 'sie war überrascht .').
Sie war überrascht.

fp(b6, 'meine schulhe liegt gegenüber meinem haus .').
Meine Schule liegt gegenüber meinem Haus.

fp(b7, 'Bastia lieht in Umbrien .').
Bastia liegt in Umbrien.
```

**Interference with English**

```plaintext
fp(c33, 'ihr name was sara .').
Ihr Name war Sara.

fp(c2, 'heisst die person links under Karin ?').
Heißt die Person links unter Karin?

fp(c4, 'mein father heisst Franco und mein brother Marco .').
Mein Vater heißt Franco und mein Bruder Marco.
```

**Interference with Italian**

```plaintext
fp(d4, 'diese, abend, darf, ich, nicht, in, die, disco, gehen .').
Diesen Abend darf ich nicht in die Disco gehen.

fp(d6, 'ich, stelle, den, kassettenrecorder, auf, den, balkon .').
Ich stelle den Kassettenrecorder auf den Balkon.

fp(d10, 'die lehrerin ist ab und zu sehr lunatisch .').
Die Lehrerin ist ab und zu sehr launisch.
```

### 3.3 Lexical errors

Lexical problems are related mainly to the selection of the appropriate preposition, which can depend on the verb (Ergänzung) or on noun semantics (Angabe). Prepositions can be omitted by a simplification mechanism or wrongly chosen for a similar words sound structure with the mother tongue (or another foreign language acquired). Italian native speakers make many errors in the use of prepositions or the verbal particle 'zu', which marks infinitive clauses.

```plaintext
Error selection of the prepositions

fp(e2, 'er verschwendet das geld an alkohol .').
Er verschwendet das Geld mit Alkohol.

fp(e3, 'sie wollte nach berlin studieren .').
Sie wollte in Berlin studieren.

fp(e5, 'sie gingen auf die bibilothek .').
Sie gingen in die Bibliothek.
```

**Error selection of the verbal particle 'zu'**

```plaintext
fp(f1, 'sie entschied sich nach muenchen fahren .').
Sie entschied sich nach München zu fahren.

fp(f2, 'sie ging in die kantine zu essen .').
Sie ging in die Kantine essen.

fp(f4, 'sie entschied nach berlin zu zurueckfahren .').
Sie entschied nach Berlin zurückzufahren.
```

**Wrong choice of auxiliary verb**

```plaintext
fp(g1, 'meine alte tante hat oft zu uns gekommen .').
Meine alte Tante ist oft zu uns gekommen.

fp(g2, 'ich habe abgefahren .').
Ich bin abgefahren.

fp(g5, 'ich habe angekommen .').
Ich bin angekommen.
```

**Unsuitable semantics**

```plaintext
fp(h1, 'wo machen sie die fremdsprachen ?').
Wo lernen sie Fremdsprachen?

fp(h2, 'wo machen sie die fremdsprachen ?').
Wir möchten Tennis spielen. Wir spielten Tennis.

fp(h3, 'mein bruder ist sehr gut ski .').
Mein Bruder kann sehr gut Ski fahren.
```

### 3.4 Morphology

Errors at the morphologic level reveal the lack of information in the use of paradigms and declinations, which are essential in the majority of inflected categories – noun, adjective, article, determiners, auxiliaries, and verbs – and are essential in the choice of a semantic feature, e.g. case creation. In this connection, congruence errors are the most frequently observed. Learners make mistakes when choosing case, genre, or number of a word. The creation of past tense forms seems also to be very problematic: learners tend to overgeneralize the rules to form past tense of weak verbs with strong verbs.

```plaintext
Congruence errors (nouns, pronouns, articles, adjectives)

fp(i1, 'nachdem er viele anzeige mit stellenangeboten gelesen hatte .').
Nachdem er viele Anzeigen mit Stellenangeboten gelesen hatte.

fp(i2, 'er wohnte mit seinen eltern in dem dorf .').
Er wohnte mit seinen Eltern in dem Dorf.

fp(j29, 'sie organisierten ihre reise .').
Sie organisierten ihre Reise.
```
3.5 Syntax

The main problem for German learners is syntax: Italian and German both have a free word order, yet the underlying sequence of sentence constituents do not coincide. Italian is an SVO (Subject-Verb-Object) language, while German is a SOV (Subject-Object-Verb) one. In addition, German positions finite verb always in the second place (V2-rule), except for subordinate clauses, in which the complementizer occupies that place and therefore the verb has to be realized in the last place of the clause. Italian native speakers tend to violate the German syntactic structure, especially the V2-Rule. With regard to negation, grammatical rules are really diverging between the two languages: German negation follows general complements and nominal phrases and goes before complements of time, place, manner and predicative and adjectival phrases. However, Italian native speakers have acquired a pre-verbal negation and thus they tend to apply this pre-verbal negation to German (see [27]).

Unsuitable usage of separable and inseparable verbs
fp(q30,'er teilnahm an diesem italienischen fernsehprogramm .').
Er nahm an diesem italienischen Fernsehprogramm teil.

Wrong use of V2-rule
fp(n2,'aus diesem grund monika half petra .').
Aus diesem Grund half Monika Petra.

Lack of subject pronoun or subject pro-drop
fp(r1,'gestern bin gegangen ins kino .').
Gestern bin ich ins Kino gegangen.

4. GI-Tutor Feedback

Learners obtain a message indicating errors and, if necessary, a description of these errors too (see [15;16;17;24;29;32]). Thereby they can recognize their language weaknesses and look for appropriate practice to improve their German.

Heiſt (see [15]) demonstrates that meta-linguistic feedback is meaningful, helpful and useful, especially if the error is
highlighted in the output. For this reason, and because it is
correct to human-computer interaction, the meta-linguistic
feedback has been chosen as the most suitable for these
exercises. We opt for immediate error feedback in order to be
effective (see [21]), and “concise” and “precise” to be
certainly checked by the learners (see [38]).

The parser is based on a fixed order of rules where errors
have to be identified in order to verify the conditions of well-
formedness. Afterwards the system produces a simple
feedback. These messages point out the errors, only some are
defined, e.g. subject-verb agreement, creation of the case,
inversion of direct and indirect object, position of negation,
constituent internal agreement, etc.

The system does not give corrections, since learners need a
simple error message in order to revise and correct the
sentence autonomously.

5. A Preliminary Evaluation

We decided to make a preliminary evaluation by comparing
two open source parsers of German with GI-Tutor using the
database of sentences we have collected. The first one is
Stanford Parser that can be found here,
http://nlp.stanford.edu/software/lex-parser.shtml and as
indicated in the entry webpage, is statistical parser that uses a
model based on Negra corpus. And an on-line parser of
German – ParZu (see [35;36]), which can be found here,
http://kitt.ifi.uzh.ch/kitt/parzu/. Both Stanford Parser and
ParZu are dependency parser and have no didactic purpose,
that is they are not conceived with a function as language-
learning tutors. They do not give any feedbacks and,
compared to GI-Tutor, they both tend to fail with an ill
formed input. On the contrary, GI-Tutor has a didactic
purpose and so it generates a feedback which can be useful
for the students to identify their errors and learn from their
corrections autonomously.

Criteria for establishing the wrongness of the analysis are
twofold: a. tags are ill chosen (a wrong label can compromise
the correct structure); b. the structure of the sentence is
partially or entirely wrong depicted; c. both tags and structure
of the sentence are erroneous. Whereas the correctness of the
analysis is proved when: a. all tags are accurately assigned; b.
both tags and structure of the sentence are adequate.

The analysis does not include any feedbacks, as neither
Stanford Parser nor ParZu check errors, nor generate
messages. They both require capitalization, and are very
sensitive on the presence of correct punctuation; but they are
unable to take the correctness of suffixes into account. With
correct sentences, they completes the analysis predominantly
right, yet in particular cases (e.g. congruence errors) they
tend to display a dependency forest (see [20]). On the
contrary, as GI-Tutor was conceived and implemented with a
didactic purpose, it has to analyse learners’ sentences,
identifying possible errors and giving a feedback. In order to
complete the analysis despite the errors, the parser accepts
both Italian and German syntactic constituent structures.

Also orthographic errors (not lower case) may influence the
outcome of the analysis: wrongly written words may receive
a wrong tag, whenever the input word is a homograph and
belongs to different grammatical categories. For this reason,
we use a backoff strategy in order to tag as noun every word
which is nonexistent in the lexicon. A message would be
displayed whenever a word is misspelled or non-existent. For
that purpose we also use a list of 1,700,000 words of German
available on-line here
http://sourceforge.net/projects/germandict/files/. This list
allows us to check whether a word actually exists in German
but is not contained in our lexica. In that case we activate a

simple morphological guesser to subtract possible suffixes of
German and look for the root.

The parser has showed some difficulties on the correct
detection of relative pronouns, sometimes tagged as
determinate articles, and coordinated constituents.

We report here below figures and tables containing results of
our evaluation organized by types of errors. We start by
looking at parse of Wrong Sentences and then move to parse
of Correct Sentence, with an overall table with all results.

As can be noticed from the Figure 1., ParZu has a better
performance in sentences containing lexical errors. Whereas
GI-Tutor has a better result in sentences containing
orthographic, morphological and syntactic errors.

![Figure 1. Correct Analyses of Wrong Sentences](image)

<table>
<thead>
<tr>
<th></th>
<th>GI-Tutor</th>
<th>ParZu</th>
<th>Stanford Parser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orthographic errors</td>
<td>85.3%</td>
<td>64.0%</td>
<td>68.0%</td>
</tr>
<tr>
<td>Morphologic errors</td>
<td>80.4%</td>
<td>63.7%</td>
<td>85.8%</td>
</tr>
<tr>
<td>Lexical errors</td>
<td>82.7%</td>
<td>88.3%</td>
<td>85.8%</td>
</tr>
<tr>
<td>Syntactic errors</td>
<td>69.7%</td>
<td>66.0%</td>
<td>60.9%</td>
</tr>
</tbody>
</table>

Table 1. Correct Analyses of Wrong Sentences

As Figure 2. below clearly shows, when parsing correct
sentences, ParZu has best results in fields of morphology,
lexic and syntax, in particular when lexical choices are
considered (98.91%).

![Figure 2. Correct Analyses of Correct Sentences](image)
When considering all sentences, GI-Tutor obtains a better performance when orthography (88.66% vs 77.33%) and morphology (83.19% vs 80.31%) are considered. In the lexical fields ParZu clearly prevails (90.76% vs 80.45%). When compared to Stanford Parser, we see again that GI-Tutor has better performance in orthography related sentences (88.66% vs 77.33%) and syntax, (73.95% vs 70.56%), but then Stanford prevails in lexical related sentences (86.95% vs 80.45%) and in morphology (87.40% vs 83.19%). The average of all results shows similar values: GI-Tutor completed a correct analysis for 81.56% of the sentences, while ParZu for 81.90% and Stanford for 80.56%.

6. Conclusion and Future Work

We tested GI-Tutor ability to parse the 1262 sentences of our database and compared the results with ParZu - Zürich open source parser of German (see [35;36]) - performance. Only 35 sentences out of 1262 received a partial output or no output, just a partial one, with no feedback at the end: this corresponds to 97.73% of accuracy. As to ParZu, 27% of our wrong sentences were not completely parsed or not parsed at all, i.e. it only got 73% accuracy. The same applies to Stanford Parser. In the case of GI-Tutor, it is worth while reminding that being a rule-based parser, it will undergo further improvements both at tagging level and at rule level before making it available for a public release.

As regards its usage for didactic purposes, more study of student-written sentences is needed in order to give a clearer overall picture of the kinds of materials the parser needs to be able to deal with. We intend to produce additional evidence based on students’ materials in the near future.

GI-Tutor will be made available on the web of the University Ca’Foscari next academic year. It is currently implemented as stand-alone application and sits on computers of our Laboratory of Computational Linguistics. We do not have yet a statistics of the usage of the system - with what students and with what results, but will try to collect it as soon as possible.

Not all the structures are correctly analysed and not all feedback is properly organized as said above. Given the extension of the task comprising all possible items of the grammar of German, we left some less common mistake still to be elaborated: this includes complex passive verbal constructions as well as extrapositions at complex sentence level.

References
