Language Technology for Automatic Control in eLearning Tools - Translator and Second Language Training -

Susanne Preuß
Institute for Applied Information Sciences
(IAI)
Martin-Luther-Str. 14
D-66 111 Saarbrücken, Germany

Christoph Rösener, Paul Schmidt
University of Saarland
Campus
D-66 123 Saarbrücken, Germany

1. Introduction

The paper introduces the results of several projects (ALLES (Advanced Long distance Education System)[1], ILLU (Internet-Lehr-Lernmodule für die Übersetzer- und Dolmetscherausbildung)[2] AUTOLEARN (AUTOmatic tutor for lifelong language LEARNing)[3], and the soon starting COMENIUS project ICE3 'Integrating CALL in Early Education Environments')[4] addressed in [2], [3], [5], [6] whose objective was and is to create e-Learning sites for students of translation and second language learners realised as autonomous learning systems. 'Autonomous learning' means that the student can work with the system on the computer without any support by a human tutor. The innovation in all these projects is that intelligent (!) automatic correction is provided for students' input. 'Automatic correction' could be seen on a trivial level on the basis of a simple matching of characters, e.g. as a gap filling exercise: 'Fill in the right preposition'. This is actually the kind of IT-based language learning at the moment. Intelligent correction as provided in our systems goes far beyond that. It is meant to be the (automatic) detection of the kind of error that has been committed by the student and the issuing of a pedagogically useful diagnosis of the kind of error. This kind of correction is provided on different levels, orthographic, syntactic and even on a content level. Our automatic correction tools (AC tools) are based on sophisticated language technology.

Automatic correction tools are made by computational linguists who develop modules for error detection on the basis of linguistic analyses. As the adaptation of sophisticated language technology to new applications or new learning units usually requires these specialists (computational linguists configuring grammars) a serious bottleneck for the usability of such systems exists at this point. In order to avoid this bottleneck a tool was developed that allows the teacher to create her own linguistic resources (to be used by the AC tools) and thus equip her own exercises with automatic correction, i.e. to generate automatically the required automatic correction tools. This is the second and most interesting component.

The paper comes in the following sections:

- A short description of the NLP tools and what they do
- Automatic correction of second language learning
- Automatic correction of translation
- The tool for generating automatic correction tools
- A summary and assessment of the results

2. The NLP-Resources for Automatic Correction

Automatic correction (AC) tools are based on natural language processing (NLP) technology. They deliver corrections on different levels of linguistic description, grammatical and orthographic correction, semantic and pragmatic adequacy of texts by checking the content of the students' productions.

The major innovation in our tools is the application of 'automatic intelligent language processing' (NLP) to language learning. The important point is that this is done without compromising the didactic

[1] Funded under IST 2001-34246
[2] Funded by Geförderte Projekte im Rahmen der Anreizorientierung eLearning 3.0, University Saarland
[3] Funded by LifeLongLearning – 135693 – LLP -1 – 2007 -1- ES-KA3-KA3MP
[4] Funded under 510653-LLP-1-2010-1-ES-COMENIUS-CMP

concept which is task-based and communicative, not syntax or vocabulary driven. So, automatic correction is not on the level of checking whether a preposition filled into a gap is correct, but e.g. the checking if a specific communicative action has been performed correctly.

The core of the processing is a morphosyntactic and a semantic analysis of the language in which the exercises to be corrected are written. The basis of the processing is available natural language processing (NLP). This is syntactic parsing with lexical semantics here simply introduced by way of illustration in Fig 1 which shows a German noun phrase, the NP ‘vertraulicher Regierungsbericht’ (confidential government report).

vertraulicher Regierungsbericht	
Strings	Attribute-Value-Pair (partial)
"vertraulicher"	{ori=vertraulicher, ehead={g=m ,nb=sg, case=nom, infl=strong}, c=adj, deg=base, cs=a, ds=vertrauen~lich, ls=vertrauen, ss=a, lng=germ, cat=adj, case=nom, nb=sg,g=m}
"Regierungsbericht"	{ori=Regierungsbericht, ehead={g=m, nb=sg, case=nom, infl=strong}, c=noun, cs=n#n, ds=regieren~ung#berichten~IRREG, ls=regieren#berichten, ss=mass-nahme#text, lng=germ, gs=f#m, lu=regierungsbericht, ts=regierungs#bericht, t=regierung#bericht, cat=noun, case=nom, nb=sg, g=m}

Fig. 1: Syntactic analysis of German NP

Fig. 1 shows that there is morphosyntactic information represented as attribute-value-pairs: Case, number, person, category, degree. There is semantic information and there is derivational information. The semantic information e.g. is that 'Regierungsbericht' is a textual object. There is information about the components of the compound. The derivational information is that Regierung (government) is derived from 'regieren' (to govern) and 'Bericht' (report) from 'berichten' (to report). The full parse of a sentence would exhibit even more information. In addition, there would be additional information about the composition of the sentence out of phrases and also some information about syntactic function (subject, direct object) and the grammar contains additional rules that account for the errors.

The grammar contains specific error rules such as the one in Fig 2 which sketches that if a subject noun has a value for person different to the one of the finite verb an error message is issued that there is an agreement error (‘wrong person’).

```
{cat=s,error1=wrong_person} →
....
   {cat=n,pers~=P,case=nom},
....
+
....
   {cat=v,pers=P,tns=pres}
....
```

Fig. 2: Person agreement checking

A second component of the linguistic processing is a ‘comparison tool’ which takes the output of the syntactic analysis as shown in Fig 1 and compares it with the posted (analysed) translation or the stored model solutions and a list of exercise-specific errors which the teacher anticipates based on her teaching experience. While the first set of rules like those in Fig 2 are of a general nature this second set of rules are exercise specific.

Assume that an exercise is: “Please read the Wikipedia article about ‘Hagia Sophia’ and answer the question below”, where the question is “From an architecture point of view, what makes the Hagia Sophia in Istanbul so famous?” The following answers are considered acceptable:

(1)(a) The Hagia Sophia is famous for its massive dome.
(b) The old mosque is famous for its massive dome.
(c) The reputation of the Hagia Sophia is due to its massive dome.
(d) The reputation of the old mosque is due to its massive dome.

The comparison tool maps the students' input on the different solutions stored in the database for this exercise. The solutions are stored in form of feature bundles as shown in Fig 1 onto which the analysed input is mapped. If there is a discrepancy in feature values or in structure a corresponding message is issued.

The challenge for automatic correction on content level is to provide substantial flexibility (where the degree of flexibility corresponds to the degree of intelligence). The acceptable solutions must not be too limited. So, as far as (1) is concerned there should not only be 'the old mosque' but also 'the very old mosque' or 'the ancient mosque' and instead of 'its massive dome' 'its dome' or 'its mighty dome' and also 'its mighty cupola' or whatever. If we try to enumerate all these possibilities we would easily end up with hundreds or even thousands of solutions.

Based on the modelling of the four possible answers in (1), one could show a solution to the problem of hundreds of solutions, namely to model answers in four blocks, corresponding to WHO, WHAT, and WHY, and linguistic expressions such as "due to", as defined above. The possible correct block sequences then would be the following:

a) B1 *B2.A* B4 b) *B2.B* B1 B3 B4

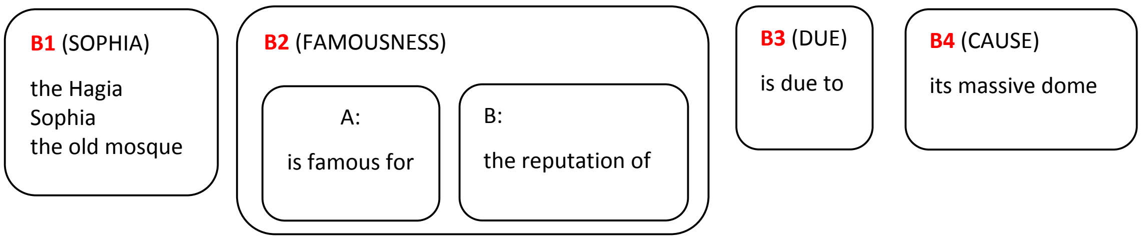
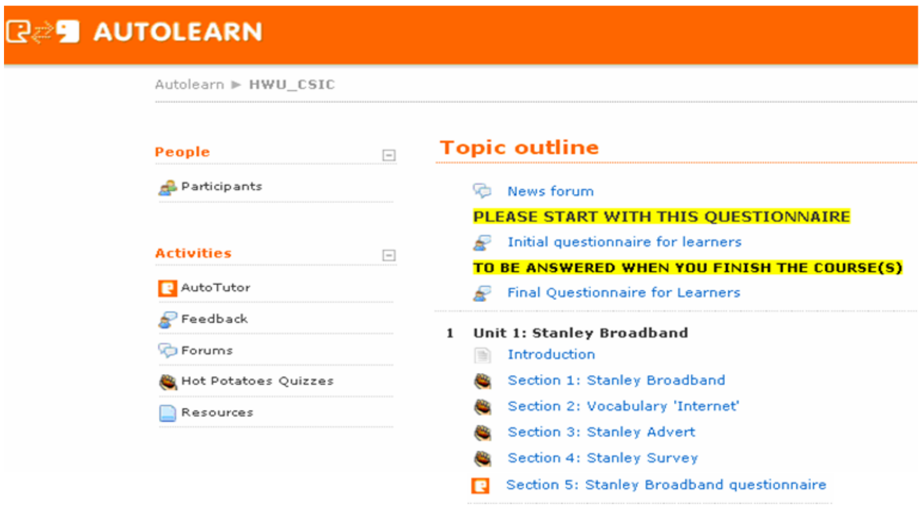


Fig. 3: Blocks as specified in AutoTutor GUI.

The innovation of this version is that it introduces a lot of flexibility as it allows for local variation without requiring the reformulation of the whole of the sentence. In addition to the greater flexibility the new strategy allows for the design of individual error handling and error messages which can now be 'block wise'. The following two sections will elaborate in more detail on how the tools are applied to different kinds of autonomous learning.

3. Automatic Correction for Language Learning

The 2nd language learning system AUTOLEARN comes as a Moodle platform. There are a set of learning units such as 'customer service and international communication' shown in Fig. 4 coming in a series of units each of which consisting of textual material and corresponding exercises. The student is supposed to go through them one by one. Those exercises with the 'AutoTutor' icon (here in section 5) are equipped with the AC tools. A typical AUTOLEARN unit on the Moodle platform looks as in Fig 4.



News forum
PLEASE START WITH THIS QUESTIONNAIRE
Initial questionnaire for learners
TO BE ANSWERED WHEN YOU FINISH THE COURSE(S)
Final Questionnaire for Learners
1 Unit 1: Stanley Broadband
Introduction
Section 1: Stanley Broadband
Section 2: Vocabulary 'Internet'
Section 3: Stanley Advert
Section 4: Stanley Survey
Section 5: Stanley Broadband questionnaire

Fig. 4: 2nd language learning on Moodle platform

In the middle column you have the lesson split into some 5 units 1 of which is shown in Fig. 4. Section 5 is an AC enabled exercise.

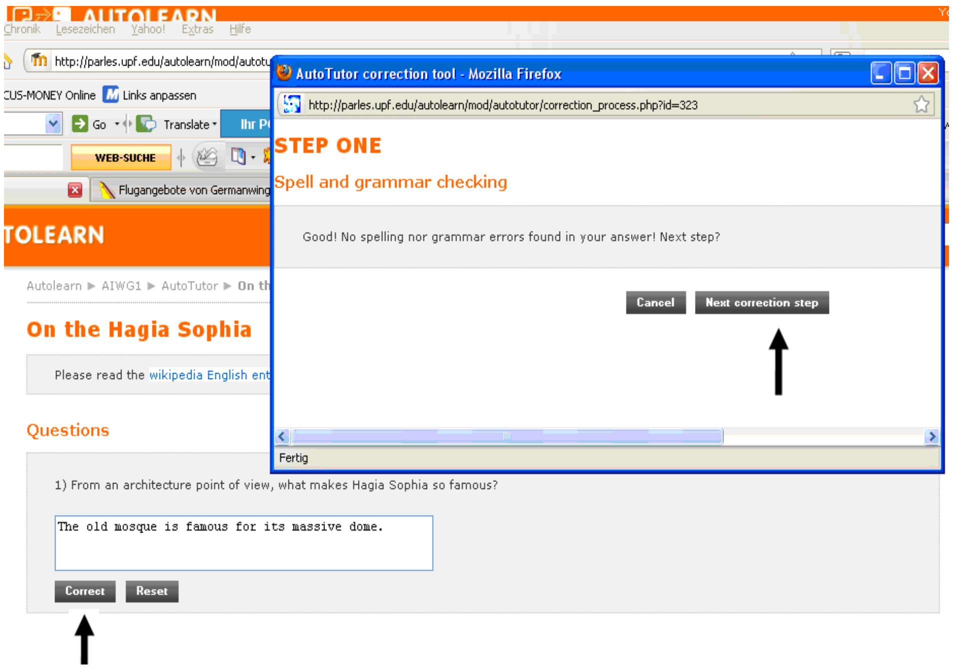
The checking comes in two steps. There is a general part of checking that concerns the general properties of language, such as syntactic and semantic correctness. There is then a specific part that concerns the correctness of the student's input as a solution to an exercise specific task.

As far as the general part of the checking is concerned there is spell checking and grammar checking which is done on a general level, so to say ‘per language in general’. Spell checking discovers non-words and is thus independent of any text sort or specific background. Grammar checking is different. Though the grammar checking also has some general rules which are checked for any text sort such as noun phrase congruence, the most important feature of the general strategy of grammatical error checking is different. It is not to make a full analysis of the input sentence on the basis of a concept of a ‘well formed sentence’ such as in classical linguistics and then calculate some distance. There is not a general formalisation of 'correct sentence of a language' and then the detection of how the student’s input deviates from this correctness. **Errors are detected directly and an appropriate error message is provided, i.e. errors are encoded, there are grammar rules that represent errors (see example in Fig 2)** .

For grammar checking it has to be taken into account which errors can be expected from the target group. E.g. they are different if a group of native speakers are target or if a group of foreign language learners are target. So, for each of the target groups specific rules are used (apart from a few that are considered general such as those that handle noun phrase congruence). On the other hand that means that everything that is not detected as erroneous is correct. The tools in AUTOLEARN detect very reliably orthographic and grammatical errors for German and English, though not all of them.

Content checking: The system is able to detect most of the correct solutions for an exercise, but probably not all possible ones. All solutions that are not known to the system as correct are marked incorrect. The system cannot make any statements about degrees of correctness as it does not ‘understand’ or interpret an input with respect to how or to which degree the solution could be considered a fulfilment of the exercise. The system can respond intelligently if it answers to a predictable error. In order to give an impression of how the system works there is an exemplification here from the AUTOLEARN project:

Fig 5 shows a page with exercises that are AC-tools enabled. There are slots where the student can put in answers to the questions one of which is shown. As soon as an answer is put in the button ‘Correct’ is to be pushed. The AC tools start working. Assume that a student has put in a sentence as in Fig 5 and pushed the button ‘Correct’.


AUTOLEARN
Chronik Lesezeichen Yahoo! Extras Hilfe
http://parles.upf.edu/autolearn/mod/autotu
Links anpassen
Go
Translate
WEB-SUCHE
Flugangebote von Germanwing
TOLEARN
Autolearn ▶ AIWG1 ▶ AutoTutor ▶ On th
AutoTutor correction tool - Mozilla Firefox
http://parles.upf.edu/autolearn/mod/autotutor/correction_process.php?id=323
Cancel
Next correction step
Fertig
Correct
Reset

STEP ONE

Spell and grammar checking

Good! No spelling nor grammar errors found in your answer! Next step?

On the Hagia Sophia

Please read the wikipedia English ent

Questions

1) From an architecture point of view, what makes Hagia Sophia so famous?

The old mosque is famous for its massive dome.

Fig. 5: Example AC tools enabled exercise

The grammar and spell checker returns the message that the input is grammatically and orthographically correct in the pop up window. This does not say anything about the appropriateness of the input as far as the content is concerned. (If we put in any other grammatically and orthographically correct sentence we would get the same feedback).

After this general checking we are now in the situation to push the 'Next correction step' button in the pop up window which triggers the content checking.

To complete the picture it has to be described how the automatic correction deals with some other cases.

One is that the input might be grammatically and orthographically incorrect in this case the orthographic error has to be corrected first as otherwise the system is not able to check according to content correctness.

Another case is that the input is grammatically and orthographically correct but nonsense as far as the content is concerned. In this case the system returns a message that the input is grammatically and orthographically correct, but issues an error message in the second step that it is nonsense (from a content point of view). The situation is a bit more difficult if the input is partially correct and partially incorrect as far as content is concerned. In this case we get a message that identifies the wrong part and says that the other parts are correct. The third case to be addressed is that it may happen that the input by the student contains the correct answer but in addition a piece of nonsense, say for the sake of the argument 'The old mosque is famous for its massive dome and the car is red'. In this case the system is able to identify the piece of nonsense and mark it as a superfluous part of the otherwise correct answer.

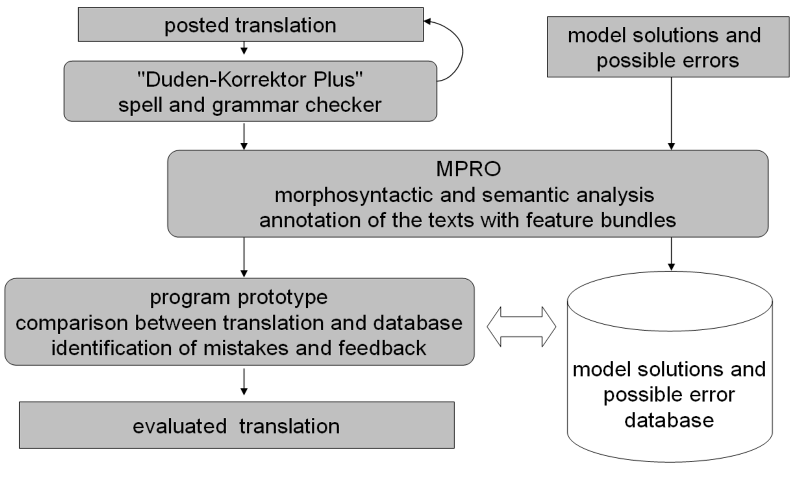
4. Automatic Correction for Teaching of Translation

The application of the tools described for automatic correction in second language learning to other scenarios is obvious[5]. So for teaching translation the first step to be done to have a tool for translators is that the instructor designs exercises probably with specific translation problems in mind. She then has to enumerate the correct solutions which can be several and exhibit a considerable degree of flexibility. In addition, the teacher gives specific erroneous examples which she expects the students might do and characterises them with an appropriate error message.

A popular translation problem English and German is present perfect structures like as in (2).

(2)(a) Er lebt seit 2 Jahren in London.
(b) He has been living in London for 2 years.
(c) *Lit.: He lives in London for 2 years

If a student gave (2)(c) as solution (which would not be a surprise for students of English with a German mother tongue) an appropriate error message is issued. The whole system is shown in fig 6.


posted translation
"Duden-Korrektor Plus" spell and grammar checker
model solutions and possible errors
MPRO morphosyntactic and semantic analysis annotation of the texts with feature bundles
program prototype comparison between translation and database identification of mistakes and feedback
model solutions and possible error database
evaluated translation

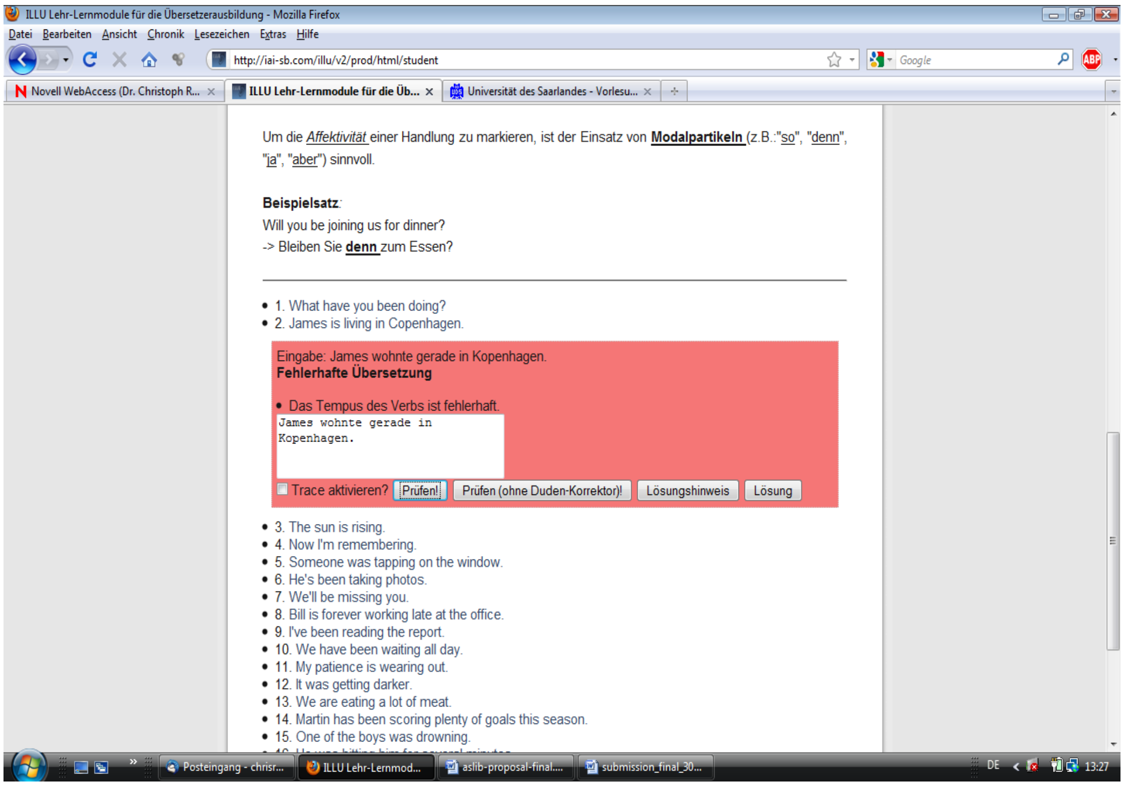
[5] In principle the tools could be used for checking ANY input involving textual material. Even economics or nuclear physics could be checked.

Fig. 6: Creation of resources and processing of input

The right column sketches the process of storing the solutions which go through the NLP tools to be analysed and then stored in the model database. The left column shows the processing chain of the students' input.

Due to the morphosyntactic and semantic analysis there are many features available for the comparative operation between the posted translation and the stored model solutions and possible mistakes. At word level the most important are the original string and the basic form, case, number, gender, tense and part of speech. On sentence level there are some more, e.g. word occurrence, word order, marking of phrases or sentences to name but a few. For the comparison operation distinct parameters were defined on the basis of which the comparison is made. The program computes whether certain feature bundles between two structures are identical or not. Depending on the various linguistic features this is done using different strategies to find the differences between the structures. Finally the various mistakes, if any, are determined and the result is sent to the next module. A differentiated definition of possible types of mistakes and their classification is one of the basic requirements of the system. Here the complexity of the error code corresponds directly with the quality of the system. The more differentiated the error code, the more powerful the system is.

The implementation of rules for the determination of mistakes is very labour-intensive at the beginning. But together with the aforementioned comparison operation these rules are responsible for the quality. The more differentiated the rules for a certain translation and the corresponding model solutions and possible mistakes, the more high-quality the system is. Beside rules based on the morphological, syntactic and semantic level (e.g. false verb, false relative pronoun etc.) it is also possible to implement rules which are sentence specific (e.g. changed constituents, word occurrence). If the topic of a certain unit is a particular translational problem, it is also possible to define specific rules for this. So far only rules on the morphological and syntactic level have been implemented. One of the ideas of the project is, that after initially collecting all rules as singular rules per text and translation, perhaps at a later date specific rules can be summarised to more abstract rules. Additionally this might be a chance to gain interesting results for translation studies.


ILLU Lehr-Lernmodule für die Übersetzerausbildung - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://iai-sb.com/illu/v2/prod/html/student
Google
Novell WebAccess (Dr. Christoph R...
ILLU Lehr-Lernmodule für die Üb...
Universität des Saarlandes - Vorlesu...
Posteingang - chrisr...
ILLU Lehr-Lernmod...
aslib-proposal-final....
submission_final_30...
DE
13:27

Um die *Affektivität* einer Handlung zu markieren, ist der Einsatz von **Modalpartikeln** (z.B.:"so", "denn", "ja", "aber") sinnvoll.

Beispielsatz:
Will you be joining us for dinner?
-> Bleiben Sie **denn** zum Essen?

- 1. What have you been doing?
- 2. James is living in Copenhagen.

Eingabe: James wohnte gerade in Kopenhagen.
Fehlerhafte Übersetzung

- Das Tempus des Verbs ist fehlerhaft.

```
James wohnte gerade in
Kopenhagen.
```

☐ Trace aktivieren? Prüfen! | Prüfen (ohne Duden-Korrektor)! | Lösungshinweis | Lösung

- 3. The sun is rising.
- 4. Now I'm remembering.
- 5. Someone was tapping on the window.
- 6. He's been taking photos.
- 7. We'll be missing you.
- 8. Bill is forever working late at the office.
- 9. I've been reading the report.
- 10. We have been waiting all day.
- 11. My patience is wearing out.
- 12. It was getting darker.
- 13. We are eating a lot of meat.
- 14. Martin has been scoring plenty of goals this season.
- 15. One of the boys was drowning.

Fig. 7: Student interface – feedback

After the translational errors have been precisely determined by the comparison operation the corresponding feedback messages are sent back to the students, as shown in Fig. 10. After processing one sentence the messages are given back to the students. Until now there is a fixed set of possible feedback messages implemented. But there is no restriction concerning the form of the feedback messages. It is for example possible to store not only detailed feedback messages for specific translational problems. In the future whole e-learning units and links to special phenomena and further literature can be provided.

5. Generation of Automatic Correction Tools: The AutoTutor Tool

During the projects described in [2] and [5] it became clear that the tools lack usability if for each extension or adaptation to a new language, course or teaching environment computational linguists would have to do the grammar writing and the modelling of the exercise specific content checking.

Having designed the 'block wise' approach as described in section 2 the idea came up that it should not be too difficult to ask a non-expert in NLP to design the blocks and describe their combination. If the user, i.e. the language teacher or the teacher of translation is provided with an easy to handle tool (a user friendly graphical interface) it could be left to her to develop her own exercises that are AC enabled.

The building blocks could be created by language teachers themselves and the rules for combining them as well, given an easy to handle interface. If the blocks are then (automatically) analysed (the same way as the full sentences described in section 2) and correctly combined we have an excellent tool for allowing the teacher to build her own exercises enabled with automatic correction. This interface was developed and is called AutoTutor.

Fig. 8 shows how the teacher's input is converted into NLP-components. Predefined system components are contained in shapes using plain borders, the resulting ones are contained in rectangles using hyphenised lines. The figure reflects the creation of answer modelling resources and error modelling resources.

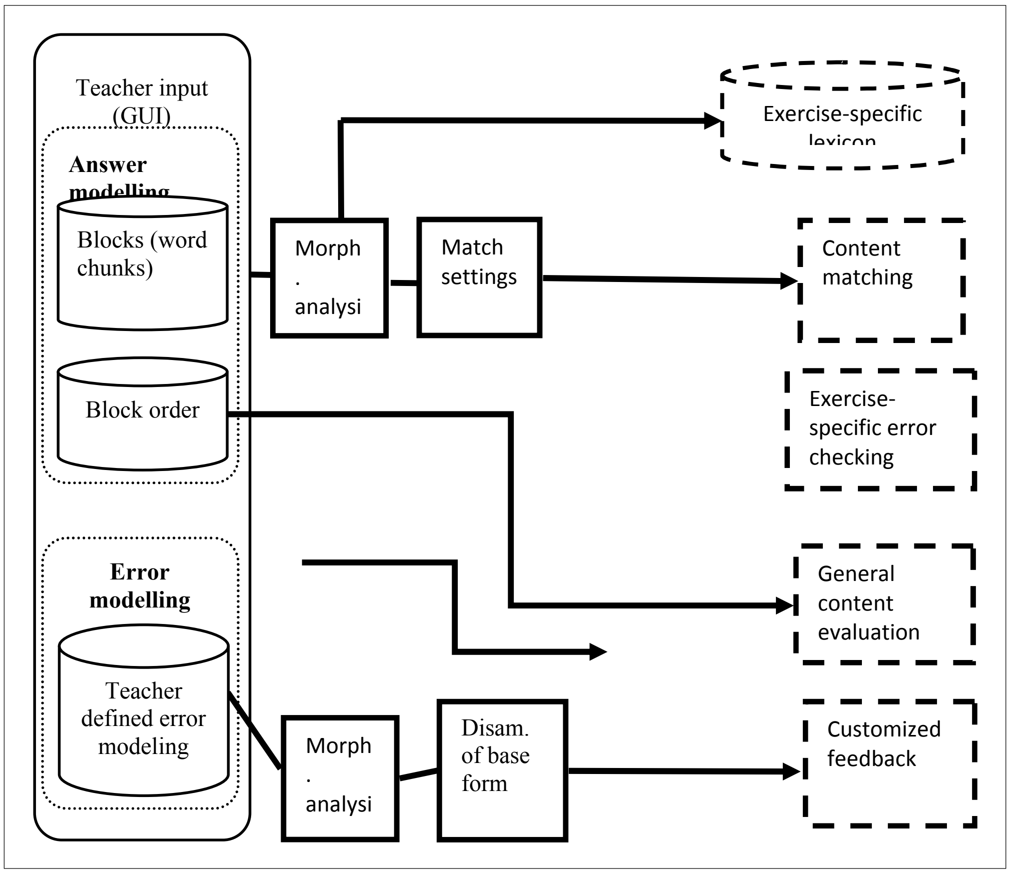
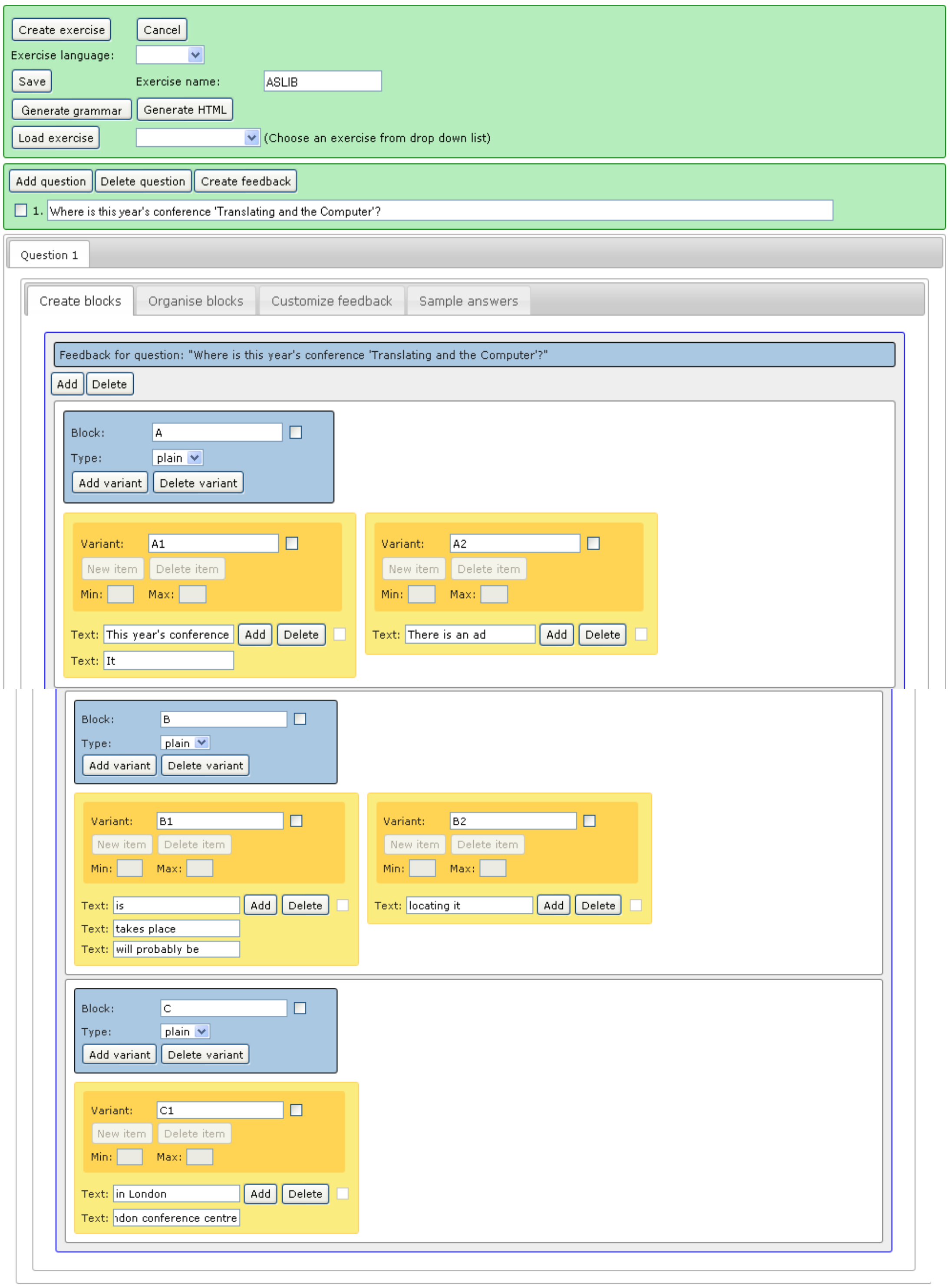


Fig 8: Processing schema and components of the customisable NLP resources

The AutoTutor tool comes as an easy to handle graphical tool which allows for an easy identification of the blocks and their combination. The following two figures are an illustration.

Fig 9 shows how blocks are created. With a few simple clicks a checking tool is created in a blockwise way. Starting out ‘Create exercise’. This opens slot ‘Add question’. The ‘create blocks’ button opens the first block (in blue). Name the block and add ‘variants’ of the block. They may represent very different structures. All variants also get a name which allows for constraining the sequence of blocks. In our example this is needed. So, block A1 can only be combined with B1 and C1, but not with B2. On the other hand block A2 can only be combined with B2 and C1, but not with B2. Finally, the button ‘Generate grammar’ compiles the grammar out and allows the tool being integrated into an exercise.



Exercise language:
Exercise name: ASLIB
(Choose an exercise from drop down list)

1. Where is this year's conference 'Translating and the Computer'?

Feedback for question: "Where is this year's conference 'Translating and the Computer'?"

Block: A
Type: plain

Variant: A1
Min: Max:
Text: This year's conference
Text: It

Variant: A2
Min: Max:
Text: There is an ad

Block: B
Type: plain

Variant: B1
Min: Max:
Text: is
Text: takes place
Text: will probably be

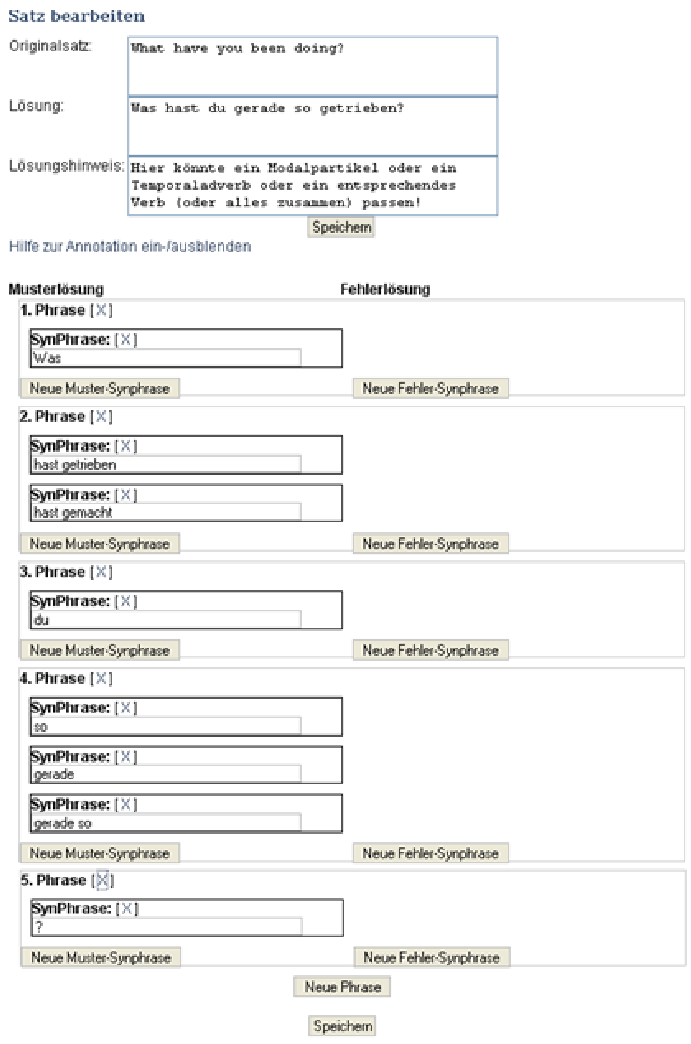
Variant: B2
Min: Max:
Text: locating it

Block: C
Type: plain

Variant: C1
Min: Max:
Text: in London
Text: ndon conference centre

Fig 9: Configuration tool for teachers of language

The interface for modelling translation tasks is very similar to the one in Fig 9. It follows the same principles. It is a web based tool that is used for teaching translation at the University of Saarland.



Satz bearbeiten

Originalsatz: What have you been doing?

Lösung: Was hast du gerade so getrieben?

Lösungshinweis: Hier könnte ein Modalpartikel oder ein Temporaladverb oder ein entsprechendes Verb (oder alles zusammen) passen!

Speichern

Hilfe zur Annotation ein-/ausblenden

Musterlösung | Fehlerlösung

1. Phrase [X]
SynPhrase: [X] Was
Neue Muster-Synphrase | Neue Fehler-Synphrase

2. Phrase [X]
SynPhrase: [X] hast getrieben
SynPhrase: [X] hast gemacht
Neue Muster-Synphrase | Neue Fehler-Synphrase

3. Phrase [X]
SynPhrase: [X] du
Neue Muster-Synphrase | Neue Fehler-Synphrase

4. Phrase [X]
SynPhrase: [X] so
SynPhrase: [X] gerade
SynPhrase: [X] gerade so
Neue Muster-Synphrase | Neue Fehler-Synphrase

5. Phrase [X]
SynPhrase: [X] ?
Neue Muster-Synphrase | Neue Fehler-Synphrase

Neue Phrase

Speichern

Fig 10: Configuration tool for teachers of translation

You also have the blockwise approach that allows for defining a great number of solutions.

6. Summary

This paper presents two sets of tools for automatic correction of students' linguistic input. The first one is for generating automatic feedback to students' input to different exercises on different levels of linguistic description, syntax, semantics, content. The second one is a tool for the teacher to create her own correction tools.

Both tools have been widely tested with hundreds of students and dozens of teachers at several universities with both applications. The acceptance by users, both students who used the correction tools as well as the teachers who used the AutoTutor tool was very good. The major innovation of the approach is that though sophisticated language technology is used for providing automatic correction a possibility has been found to allow a teacher who does not have any computational linguistic skills to automatically generate her own exercises equipped with automatic correction. After a short introduction the teachers were capable of building their own exercises with automatic correction.

A new project will distribute these tools amongst high schools. First contacts suggest that there is great interest.

Finally, it should be mentioned that the tools introduced here could be used for any kind of checking textual input even in completely different environments such as geology or medicine. The tools allow for checking any input that is available in textual form.

References

[1] Carl, Michael, and Antje Schmidt-Wigger (1998). Shallow Post Morphological Processing with KURD. In Proceedings of NeMLaP'98, Sydney.

[2] Quixal, Martí, Toni Badia, Beto Boullosa, Lourdes Díaz, and Ana Ruggia. (2006). Strategies for the Generation of Individualised Feedback in Distance Language Learning. In Proceedings of the Workshop on Language-Enabled Technology and Development and Evaluation of Robust Spoken Dialogue Systems of ECAI 2006. Riva del Garda, Italy, Sept. 2006.

[3] Quixal, Martí , Preuß, Susanne, García-Narbona David, Boullosa, Jose R. (2010): AutoTutor: a tool for teachers to design NLP-intensive CALL activities. Proceedings of NAACL, Los Angeles, 2010.

[4] MeLLANGE (2006): „WP4 – Typologie der Übersetzungsfehler“, in http://mellange eila.jussieu.fr/ Annotation_Schemes/current_translation_error_tree_de_53.jpeg [13.05.2010].

[5] Rösener, Christoph: “A linguistic intelligent system for technology enhanced learning in vocational training – the ILLU project”. In Cress, Ulrike; Dimitrova, Vania; Specht, Marcus (Eds.): Learning in the Synergy of Multiple Disciplines. 4th European Conference on Technology Enhanced Learning, EC-TEL 2009 Nice, France, September 29 – October 2, 2009 Proceedings. Lecture Notes in Computer Science. Programming and Software Engineering, Vol. 5794, 2009, XVIII, p. 813, Springer, Berlin.

[6] Schmidt, Paul, Sandrine Garnier, Mike Sharwood, Toni Badia, Lourdes Díaz, Martí Quixal, Ana Ruggia, Antonio S. Valderrabanos, Alberto J. Cruz, Enrique Torrejon, Celia Rico, Jorge Jimenez. (2004) ALLES: Integrating NLP in ICALL Applications. In Proceedings of Fourth International Conference on Language Resources and Evaluation. Lisbon, vol. VI p. 1888-1891. ISBN: 2-9517408-1-6.