Semi-automatic Test Generation for Tandem Learning

Manfred Klenner, Simon Clematide, Michi Amsler

Institute of Computational Linguistics, University of Zurich, Switzerland
{klenner,siclemat,amsler}@cl.uzh.ch

Abstract

We introduce a Web-based CALL architecture that facilitates the construction of learner-customized multiple choice tests in a cross-lingual tandem language learning environment. Mistakes made by the learner are manually corrected and classified by his tandem partner, who acts as a tutor. If the learner has problems to identify and correct his mistakes, or if he likes to practice, he can generate two kinds of tests that are automatically produced on the basis of the tutor's error classifications. We describe the NLP components behind our test generator, give a preliminary evaluation of the quality of the generated tests, and present the current state of our Web application.
Index Terms: CALL, NLP, Syntactic Analysis, Tandem Learning

1. Introduction

Web applications that support collaborative working and learning are becoming more and more popular, e.g. in form of Wikis. We currently are developing such a Web-based CALL application for tandem language learning. Here, each partner is both, learner (novice) and tutor (expert). In his role as a tutor, he corrects erroneous input of his partner and, moreover, explains the errors being made. The quality of such explanations depends on several factors, among them the explanatory skills of each partner. We believe that the following learning scenario could help to improve the quality of these – somehow critical – situations. The tutor still corrects the mistakes made by the learner, but he also classifies it according to a simple error taxonomy. This allows our system to automatically generate two kinds of multiple choice tests (described below). The learner then is confronted with his text, where sentences with mistakes are highlighted. If he is able to correct it, everything is fine. But if not, he could go through a series of automatically generated tests in order to learn more about the mistakes made by him and in order to improve his problem-solving skills. The error classification of the tutor is the basis for our test generator. Correcting and classifying not only raises the problem-solving language awareness of the tutor and learner (see [1] for an in-depth overview of this concept) but also helps to enlarge the CALL database with new material, fruitful to other tandems. We first describe the test types we have in mind and then discuss the NLP machinery behind it. We discuss a preliminary evaluation of our tests and present the state of our Web interface.

2. The Sentence Corpus

Our experiments are based on a CALL sentence corpus (cf [2]) comprising about 20'000 German sentences of Japanese second language learners[1]. We randomly selected 1000 sentences from the first year (the corpus is split into first, second and third year) and corrected and annotated 270 sentences that contained exactly one error. This way we got about 270 pairs of ill-formed sentences and their corresponding correct versions. The next step was to analyze the errors and to design a set of error messages (see section 5). We added 230 well-formed sentences to the sentence pool in order to make the classification task more challenging. The corrected versions of all sentences were parsed with a dependency parser [3]. The parse trees form the basis of our error message selection component. We now briefly describe our test types. For an older version of our model see [4]. Please note that this sentence corpus later, i.e. as soon as our tandem learner is online, will be replaced (or augmented) by new sentence pairs (ill-formed and corresponding corrected sentence) that come from the users of the platform.

3. Type I: Error Class Selection

Here an ill-formed sentence is given together with 4 error messages. The learner's task is to identify the correct error class.

Take, for instance, the following ill-formed sentence. 'Aber er hat andere Frau' (*'But he has other wife'). Here, a determiner is missing: 'Aber er hat *eine* andere Frau.' Our system generates the error messages (1-4) given in Fig. 1.

	Aber er hat andere Frau
1.	The word order is wrong.
2.	A determiner is missing.
3.	The inflection of a pre-nominal word is wrong.
4.	The sentence is correct.

Figure 1: *Error Class Selection Scenario*

The error classifications require only basic grammar vocabulary. The learner has to decide whether the sentence is correct (option 4) or not and if not, which error message describes the situation best. The design challenge here is, of course, to automatically identify good (i.e. fitting) distractors. The criteria are straightforward and can easily be checked on the basis of the parser output. For instance, there must be a pronominal word, i.e. a word that syntactically depends on a noun, in order to make message 4 a meaningful description of the potentially ill-formed sentence.

4. Type II: Error Sentence Selection

In this scenario, a error message is given together with 4 sentences[2] (exactly one is ill-formed). The learner is supposed to

[1]The type of exercise behind these sentences is unknown, we only know that they were send be email to the language teachers.

[2]In the tandem learner scenario, these 4 sentences are new to the learner, i.e. we exclude the sentences produced by the learner.

find the sentence that bears the error described by the message. See Fig. 2 for an example. Again, the challenge is to automatically identify only sentences that are high quality distractors for such an error message. Here, a preposition must be present, which can be decided by the part-of-speech (PoS) tags provided by the parser[3].

	A wrong preposition is being used.
1.	Peter will in diesen Bus einsteigen. (*Peter is going to enter this bus.*)
2.	Die Wartburg wurde im 11. Jahrhundert errichtet. (*The Wartburg was build in the 11th. century.*)
3.	In Amerika bauen sie moderne Bauten. (*In America, they build modern buildings.*)
4.	Aber sie wartet an ihren Freund. (*But she waits for her friend.*)

Figure 2: *Sentence Selection Scenario*

5. The Error Classes

The tandem learners need to correct each others errors but also to classify each error according to an error classification. These two steps form the basis of our automatically generated tests. The classification is optimized for a descriptive error diagnosis based on basic grammar vocabulary[4]. Our 47 error messages attached to these error classes refer to part-of-speech tags (PoS) (e.g. adjective, noun, verb, reflexive pronoun), word order and orthography. The following two error message pattern cover 31 of the total of 47 error messages: a PoS is wrong (22 messages); a PoS is missing (9 messages).

Most of the partners in a tandem learning situation are not linguists, and thus would not accept (i.e. work with) sophisticated error classes where multiple classifications could be applied to a single error. We believe that our plain and simple error classes are well suited to enable a language learner to identify mistakes and to enable him in his role as a tutor to classify the errors of his partner.

6. Test Generation

We now turn to the question of how to determine adequate distractors for a given item, i.e. error messages (type I) or sentences (type II)? Take e.g. error message 12: “a fusion of the preposition and the determiner is needed”[5]. Of course, a preposition and determiner must be present in the sentence, otherwise the message is not appropriate. But the preposition also must immediately precede the determiner, i.e. a structural condition has to be verified.

Inverting error message 12, we get message 14, i.e. “a fusion of a preposition and a determiner is wrong”. This rule is only appropriate if such a fusion is present in the sentence. Other messages require structural relations to be checked, e.g. whether a subordinate clause is given. Most of the time, however, the presence or absence of one or more part-of-speech tags is sufficient. For example error message 2: “a reflexive pronoun is missing”. No reflexive pronoun must be in the sentence, but since auxiliary or modal verbs do not require reflexives, it also must be guaranteed that at least one main (content) verb is in the sentence.

For convenience, we are using a rule-based approach to specify these triggering conditions. The rule language in the interpreter resembles the TIGERSearch query language [5]. We have, however, adapted it to our present needs, namely to check for the presence (existence) and absence (negated existence) of tags and structural constellations. The action part is very simple, it consists of a list of error message codes applicable to the sentence in question.

In Fig. 3 we give an example of such a rule (with rule id m-12). The condition part is a conjunction of tests on part-of-speech tags or on structural relations. `X='PRPClass'` searches for a node in the syntax tree that is tagged as a preposition. X is variable that binds the id of a node for later reuse (e.g. `X<Y` which says that node X must precede Y).

```
m-12\#
    X='PRPClass' &  % X is a preposition
    Y='ART' &       % Y is a determiner
    X<Y             % X precedes Y
       ==>
       error_messages([prpart-m]).
```

Figure 3: *Rule m-12*

This rule triggers, if a preposition is immediately followed by an article. In this case a fusion is possible, prpart-m is the ID of the corresponding error message to be printed.

6.1. Empirical Evaluation

In a small-scale evaluation with 2 learners, we tested how adequate our German error classification system functions in its use for our two types of single choice tests. Test person 1 is Italian speaking and has medium German knowledge. Test person 2 is Romanian speaking and has good German knowledge. Both test persons have linguistic expertise. Each test person had to solve a number of single choice items of type I and II, and simultaneously, they had to rate the difficulty of solving the task on a scale of three values: easy, medium, difficult. Our hypothesis works as follows: If our error classification system is adequate, then our test persons should fail more tests they reckon as difficult. Or viewed from the opposite: Tests reckoned as easy by

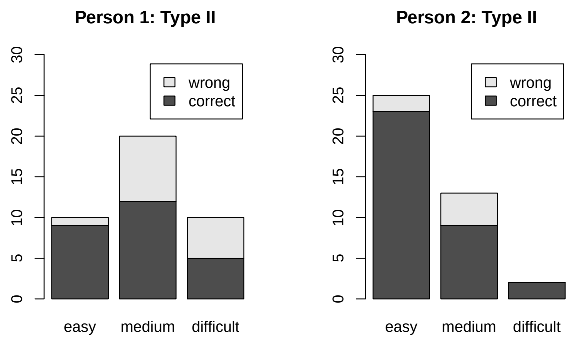


Figure 4: *Evaluation of error message selection. Correlation of correct and wrong answers with respect to self-assessment of an item.*

[3]Sentence 4 is ill-formed, the preposition should be *auf* not *an*.

[4]As opposed to a more explanatory-oriented form where deeper linguistic knowledge is needed in order to understand and apply it.

[5]In German, a preposition (e.g. *in*) and a definite, masculine or neuter determiner (e.g. *dem*) can be merged (e.g. *in dem* $\rightarrow$ *im*).

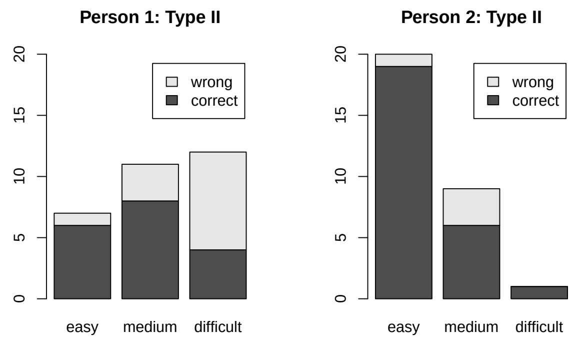


Figure 5: *Evaluation of error sentence selection. Correlation of correct and wrong answers with respect to self-assessment of an item.*

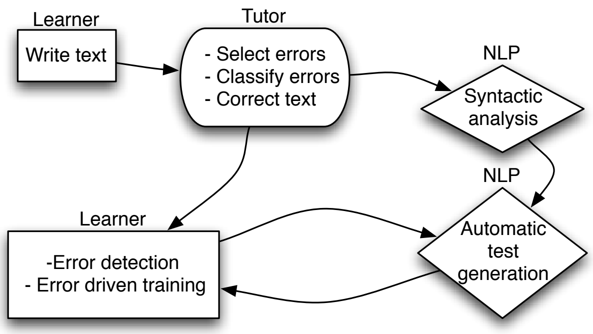


Figure 6: *Work flow of tandem learning scenario which includes natural language processing (NLP) of corrected text segments of the learner and automatic test generation driven by the learner's errors.*

the test persons should be solved correctly.

Fig. 6 and 5 show a strong correlation between the self-assessed difficulty and the error rate. Additionally, the error rate of person 2 corresponds to her advanced German skills.

7. Tandem Learning Scenario

Tandem learning is a collaborative communication effort: A second language learner Q of language L_1 is tutored by a native speaker S, and at the same time, S is tutored by Q in Q's native language L_2. In face-to-face tandem learning, the relation between the learners is reciprocal and synchronous. In an e-tandem setting with asynchronous written communication, reciprocality is not strictly necessary. For instance, language learners can post their text snippets for correction by native speakers as it is realized on the Web 2.0 online language learning platform http://www.babbel.com. We present a scenario of e-tandem learning where the principles of language awareness, viz. "‘alertness, detection and orientation"’ [1] are implemented by a Web application. We provide support for both, the tutor and the learner.

The *tutor interface* supports the correction of the learners text which includes the exact localization and classification of errors and submitting a corrected version of erroneous text segments. Fig. 7 shows the current implementation of the Web-based correction interface for tutors[6].

The *learner interface* supports alerting the learner piece by piece to mistakes in his texts, letting him detect the exact error locations and the type of error he made. The learner then decides whether he would like to train his ability to recognize the type of errors found in his texts. The system then automatically generates test items that are based on his errors, as described in the section before.

Fig. 6 shows a diagram illustrating the work flow between the different participants and processing steps in our scenario.

7.1. Representation of corrected material

We chose to store the corrected and annotated data as XML data in order to facilitate data interchange. Therefore it should be easy for any other application to use this data without any further preprocessing.

Basically our XML format consists of four parts. The first part contains optional metadata, for instance information on the native language of the learner. The second part contains the text, which can be as short as a sentence or consist of several paragraphs. The third part are the corrected erroneous text segments which contain information about the original version, the corrected version and the position in the aforementioned text. For each corrected text segment its annotated errors are stored, i.e. the selected error segments (normally the erroneous word or a text span where something is missing), the error class, an optional comment for the learner and some technical positional information.

[6]See http://kitt.cl.uzh.ch/kitt/icall/.

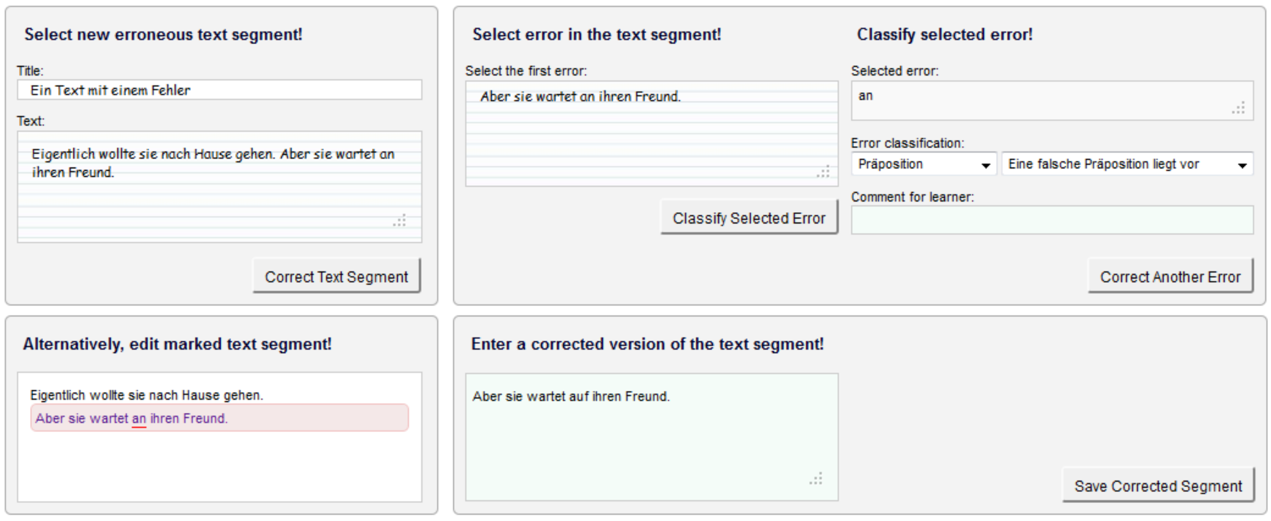
8. Related Work

To the best of our knowledge, [6] was the first who discussed "learning from errors" (‘Aus Fehlern lernen’) as a CALL setting. We have adopted his idea, but have made it fully operational. Heringer's original work relies exclusively on a manual encoding of exercises. Moreover, we designed a second exercise type (Sentence Selection) that is based on the same techniques and resources.

For German, [7] provides a linguistically fine-grained taxonomy with 157 different error diagnostics for his Augsburger error corpus (circa 7000 errors). The taxonomy is hierarchically organized and the top categories partially consist of part-of-speech categories as in our case. But he also includes top categories as ellipsis (missing elements), "plus" (obsolete elements), agreement, and morphological errors that are related to part-of-speech information on a more fine-grained levels. This classification scheme produces a lot of diagnostic instances where more than one classification is possible. This, and the large number of error messages, prevent it from being useful in a scenario where mainly non-linguists are to be expected.

G. Faass [2] originally collected the Japanese second language learner sentence corpus that we are using in our work. In her master thesis she described a CALL system that is able to parse ill-formed sentences and to diagnose the assumed errors. Her system is based on the work done by Fortmann & Forst [8], a LFG-based grammar checker.

There are various approaches that are complementary to our system. [9] describes how an annotated learner corpus (such as ours) could be used to detect errors automatically. On the other hand, if no training data is available, methods from machine translation could help (cf. [10]). Finally, [11] examines how


Select new erroneous text segment!

Title: Ein Text mit einem Fehler

Text: Eigentlich wollte sie nach Hause gehen. Aber sie wartet an ihren Freund.

Select error in the text segment!

Select the first error: Aber sie wartet an ihren Freund.

Classify selected error!

Selected error: an

Error classification: Präposition | Eine falsche Präposition liegt vor

Comment for learner:

Alternatively, edit marked text segment!

Eigentlich wollte sie nach Hause gehen.
Aber sie wartet an ihren Freund.

Enter a corrected version of the text segment!

Aber sie wartet auf ihren Freund.

Figure 7: *Screenshot of our Web-based correction interface for tutors. The upper left panel allows the selection of new text segments for correction. The lower left panel shows already corrected segments (rounded box with light red background) with exact error locations underlined red. The upper right panel permits to pinpoint and classify up to 3 errors in a segment. The error classification is two step: First a basic classification is selected, then depending on the basic classification more fine-grained error diagnostics are offered to the tutor. Optional comments serve as clarifications or background information for the learner. In the lower right panel the tutor is expected to enter a correct version of the erroneous text segment.*

an annotated learner corpus can be used to even correct faulty input. A good survey of the state of the art in intelligent CALL is the book of [12].

9. Conclusion

The use of Web 2.0 and social media for foreign language learning gained a lot of attraction in the last years [13]. The traditional Web has already been used as an environment for tandem language learning [14]. We are currently devolping a NLP-based Web 2.0 tandem language learning platform with learner-centered test generation. The tandem learners correct and classify the mistakes of their partners, thereby also increasing the underlying sentence corpus. Our test generator draws on NLP techniques such a part-of-speech tagging and syntactic parsing. The first component of the Web interface, the tutor interface, is already finished. The next step is the learner interface, with the integration of our (fully operational) test generator. We then start using it with real students. Parallel to these efforts, we are going to expand our test generator to a second language in order to support both sides of the cross-lingual tandem.

We believe that there are other useful forms of automatically generated learning material (e.g. tests) based on NLP techniques. One only has to create scenarios – as ours –, where the brittleness of NLP techniques has no (or only little) influence on the quality of the resulting learning scenario.

10. References

[1] A. M.-L. Svalberg, "Language awareness and language learning," *Language Teaching*, vol. 40, no. 04, pp. 287–308, 2009.

[2] G. Faas, "Eine LFG-Lernergrammatik für japanische DeutschlernerInnen," Universität Stuttgart, Institut für maschinelle Sprachverarbeitung, Diplomarbeit Nr. 30, Tech. Rep., 2005.

[3] K. Foth, W. Menzel, and I. Schröder, "Robust parsing with weighted constraints," in *Natural Language Engineering, 11(1)*, 2005, pp. 1–25.

[4] M. Klenner, S. Clematide, and B. Peric, "What (the hell) is wrong? An approach to semi-automatic construction of self correction tests," in *Proc. of the Workshop on NLP for Educational Resources. In conjunction with RANLP07*, 2007, pp. 15–22.

[5] W. Lezius, "TIGERSearch - ein Suchwerkzeug für Baumbanken," in *Proc. der Konferenz zur Verarbeitung natürlicher Sprache (KONVENS)*, 2002.

[6] H. J. Heringer, "Aus Fehlern lernen," http://www.philhist.uni-augsburg.de/lehrstuehle/germanistik/DaF/projekte/fehler, 1995.

[7] ——, "Fehlertypologie," http://www.philhist.uni-augsburg.de/faecher/germanis/daf/neu/fehler/f-typ.pdf, 2007.

[8] C. Fortmann and M. Forst, "A LFG grammar checker for CALL," in *Proc. of the InSTIL/ICALL Symposium, NLP and Speech Technologies in Advanced Language Learning Systems, Venice, Italy, 17-19 June*, 2004.

[9] E. Izumi, K. Uchimoto, T. Saiga, T. Supnithi, and H. Isahara, "Automatic error detection in the Japanese learners' Englisch spoken data," in *Proc. of the 41st Annual Meeting of the ACL*, 2003, pp. 145–148.

[10] J. Lee, M. Zhou, and X. Liu, "Detection of non-native sentences using machine-translated training data," in *Proc. of NAACL-HLT, Compagnion Volume*, 2007, pp. 93–96.

[11] C. Brockett, B. D. William, and M. Gamon, "Correcting ESL errors using phrasal SMT techniques," in *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the ACL*, 2006, pp. 249–256.

[12] T. Heift and M. Schulze, *Errors and Intelligence in Computer-Assisted Language Learning*. New York: Routledge, 2007.

[13] M. Thomas, Ed., *Handbook of Research on Web 2.0 and Second Language Learning*. Hershey, New York: Information Science Reference, 2009.

[14] C. Appel and T. Mullen, "Pedagogical considerations for a web-based tandem language learning environment," *Computers and Education*, vol. 34(3-4, pp. 291–308, 2000.