Generating Feedback for English Foreign Language Exercises

Björn Rudzewitz **Ramon Ziai**
Kordula De Kuthy **Verena Möller** **Florian Nuxoll** **Detmar Meurers**

Collaborative Research Center 833
Department of Linguistics, ICALL Research Group*
LEAD Graduate School & Research Network
University of Tübingen

Abstract

While immediate feedback on learner language is often discussed in the Second Language Acquisition literature (e.g., Mackey 2006), few systems used in real-life educational settings provide helpful, metalinguistic feedback to learners.

In this paper, we present a novel approach leveraging task information to generate the expected range of well-formed and ill-formed variability in learner answers along with the required diagnosis and feedback. We combine this offline generation approach with an online component that matches the actual student answers against the pre-computed hypotheses.

The results obtained for a set of 33 thousand answers of 7th grade German high school students learning English show that the approach successfully covers frequent answer patterns. At the same time, paraphrases and meaning errors require a more flexible alignment approach, for which we are planning to complement the method with the CoMiC approach successfully used for the analysis of reading comprehension answers (Meurers et al., 2011).

1 Introduction

In Second Language Acquisition research and Foreign Language Teaching and Learning practice, the importance of individualized, immediate feedback on learner production for learner proficiency development has long been emphasized (e.g., Mackey 2006). In the classroom, the teacher is generally the only source of reliable, accurate feedback available to students, which poses a well-known practical problem: in a class of 30 students, with substantial individual differences warranting individual feedback to students, it is highly challenging for a teacher to provide feedback in class or, in a timely fashion, on homework.

Intelligent Language Tutoring Systems (ILTS) are one possible means of addressing this problem. For form-focused feedback, ILTS have traditionally relied on online processing of learner language (Heift and Schulze, 2007; Meurers, 2012). They model ill-formed variation either explicitly via so-called mal-rules (e.g., Schneider and McCoy 1998) or by allowing for violations in the language system using a constraint relaxation mechanism (e.g., L'Haire and Faltin 2003).

One problem with such approaches is that they do not take into account what the learner was trying to do with the language they wrote, e.g., which task or exercise they were trying to complete. Yet the potential well-formed and ill-formed variability exhibited by learner language can lead to vast search spaces so that integrating top-down, task information is particularly relevant for obtaining valid interpretations of learner language (Meurers, 2015; Meurers and Dickinson, 2017). Given that incorrect feedback is highly problematic for language learners, ensuring valid interpretations is particularly important. Combining the bottom-up analysis of learner data with top-down expectations, such as those that can be derived from an exercise being completed, can also be relevant for obtaining efficient processing.

In this paper, we present an approach that pursues this idea of integrating task-based information into the analysis of learner language by combining offline hypothesis generation based on the exercise with online answer analysis in order to provide immediate and reliable form-focused feedback. Basing our approach on curricular demands and the exercise properties resulting from these demands, we generate the space of well-formed and ill-formed variability expected of the learner answers, using the well-formed target answers provided for the exercises as a starting point. We thus avoid the problems introduced by directly

[*] http://icall-research.de

analyzing potentially ill-formed learner language. Since generation is done ahead of time, before learners actually interact with the system, we also avoid the performance bottleneck associated with creating and exploring the full search space at run time. The resulting system can be precise and fast in providing feedback on the grammar concepts in a curriculum underlying a given set of exercises.

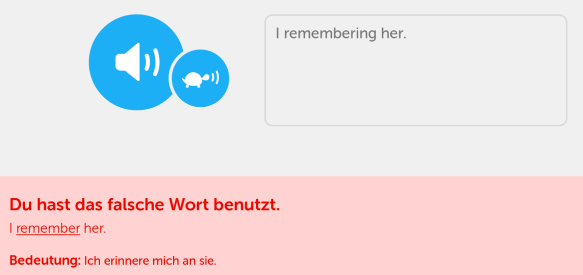
The paper is organized as follows: Section 2 discusses relevant related work before section 3 introduces our system and section 4 provides an overview on the data we elicit. In section 5, we dive into the feedback architecture and explain both the offline and online component of the mechanism in detail. Section 6 then provides both a quantitative and a qualitative evaluation before section 7 concludes the paper.

2 Related Work

Intelligent Language Tutoring Systems (ILTS) proposed in the literature range from highly ambitious conversation machines (e.g., DeSmedt 1995) to more modest workbook-like approaches (e.g., Heift 2003; Nagata 2002; Amaral and Meurers 2011). However, as discussed by Heift and Schulze (2007), the vast majority of the systems are research prototypes that have never seen real-life testing or use. We therefore limit our discussion here primarily to practical systems that are in use for foreign language learning.

In the domain of general-purpose tools, there are a number of writing aids and grammar checkers available, such as *Grammarly* (http://grammarly.com) and *LanguageTool* (http://languagetool.org). They offer grammar and spelling error correction for arbitrary English text and are intended to assist (non-native) writers of English in composing texts. Such general-purpose systems do not have any information on what the writer is trying to accomplish with the text. As a result, while local grammatical problems such as subject-verb agreement are well-within reach for such tools, the identification of contextually inappropriate forms, such as wrong tense use in a narrative, require task information.

One step further in the direction of task-based language learning, one finds tools such as *duolingo* (von Ahn, 2013). *duolingo* offers exercises for learners of various languages, mainly based on translation into or from the target language. Learners can input free-text answers and obtain immediate feedback from the system. However, while for certain phenomena the feedback is quite explicit and accurate (Settles and Meeder, 2016, p. 1849), cases such as the one in Figure 1 are not handled appropriately.



I remembering her.

Du hast das falsche Wort benutzt.

I remember her.

Bedeutung: Ich erinnere mich an sie.

Figure 1: Problematic feedback in duolingo

The learner used the *-ing*-form of the verb *to remember* in place of the simple present. Instead of identifying the form and recognizing that the lemma is the same as that in the expected answer, *duolingo* responds with ‘You used the wrong word’, which is misleading the learner to select another word. For more appropriate feedback, more metalinguistic information about the identified and the expected form would be needed. However, manually specifying such information quickly becomes infeasible even for relatively closed task types, as shown by Nagata (2009, p. 563) in the context of the *Robo-Sensei* system.

Laarmann-Quante (2016) proposes an approach for the diagnosis of spelling errors in the writing of German children that was independently developed but is conceptually similar to the perspective we pursue in this paper. Instead of attempting to process the erroneous forms directly, Laarmann-Quante obtains phonological analyses for correct spellings and uses rewrite rules that emulate typical misspellings to derive alternatives that can then be matched against actual input. However, the approach is limited to spelling errors and relies heavily on a model of German orthography. It does not target other linguistic levels of analysis, such as morphology and syntax, and the potential interaction of well-formed and ill-formed variability at the sentence level.

3 The Tutoring System

The feedback mechanism discussed in this article is implemented as part of a web-based online workbook *FeedBook* (Rudzewitz et al., 2017; Meurers et al., 2018). The foreign language tutoring system is an adaptation of a paper workbook for a 7th grade English textbook approved for use in German high schools. The FeedBook provides an interface for students to select and work on exercises. For exercises that aim at teaching grammar topics, students receive automatic, immediate feedback by the system informing them whether their answer is correct (via a green check mark) or *why* their answer is incorrect (via red color, highlighting of the error span, and a metalinguistic feedback message). The message is formulated as scaffolding feedback, intended to guide the learner towards the solution, without giving it away. The process of entering an answer and receiving feedback can be repeated, incrementally leading the student to the correct answer. If there are multiple errors in a learner response, the system presents the feedback one at a time.

Students can save and resume work, interact with the system to receive automatic feedback and revise their answers, and eventually submit their final solutions to the teacher. In case the answers are all correct in a selected exercise, the system grades the submission automatically, requiring no work by the teacher. For those answers that are not correct with respect to a given target answer, the teacher can manually annotate the with feedback parallel to the traditional process with a paper workbook. Any such manual feedback is saved in a feedback memory and suggested automatically to the teacher in case the form occurs in another learner response to this exercise. The system provides students with immediate feedback in circumstances where they would normally not receive it, or only after long delay needed for collecting and manually marking up homework assignments, while at the same time relieving teachers from very repetitive and time-consuming work. The exercises are embedded in a full web application with a messaging system for communication, a profile management including e-mail settings, tutorials for using the system, classroom management, and various functions orthogonal to the NLP-related issues (cf. Rudzewitz et al., 2017).

4 Elicited Data

The FeedBook system is being used since October 2016 in several German secondary schools as part of the regular 7th grade English curriculum. The data analysis discussed here is based on a March 2018 snapshot of the data. We collected 6341 submissions of complete exercises by 538 7th grade students from whom we received written permission to use their data in pseudonymized form for research.

From the total of 234 tasks implemented in the system, in the current system version 111 provide the immediate feedback that is introduced and evaluated in this paper. The feedback-enabled tasks include 64 short answer tasks (usually one sentence as input) and 47 fill-in-the-blanks tasks (usually one word to one phrase as input).

The frequency distribution in Figure 2 shows the number of submissions (y-axis) per task in the system, ranked from most frequent to least frequent (x-axis). Blue bars denote that the task provides immediate feedback, and yellow bars indicate that the system does not provide any automatic feedback (these are the tasks where the teacher can manually provide feedback through the system). The figure shows a tendency that more submissions exist for tasks that provide immediate feedback: out of the top 50 most worked on tasks, 36 of them (72%) provide immediate feedback. These 36 tasks are balanced between 17 fill-in-the-blanks and 19 short answer tasks.

Each submission for a feedback-enabled task provides an interaction log that stores intermediate answers and the feedback that the system provided to each answer. In section 6, we use these intermediate answers in an evaluation of the feedback approach, after introducing the architecture in the next section.

5 Feedback Architecture

In this section, we describe the feedback mechanism implemented as part of the tutoring system. The main idea behind our approach is that identifying the well-formed and ill-formed variability of possible learner answers elicited by different tasks is the key to providing precise feedback. Our feedback mechanism thus relies on well-formed target answers available for each task and generates hypothesis about possible learner answers on the basis of these target answers. This is a key difference to the use of traditional mal-rules, which operate on learner language and thus need to analyze the potentially ill-formed interlanguage of students: instead of trying to model learner language, we start from the standard, native language, for which most computational linguistic models have been developed.

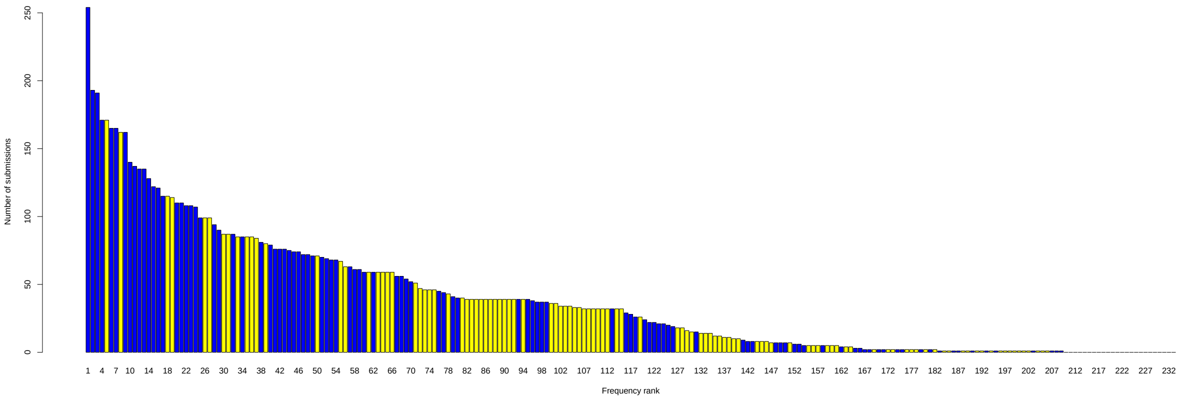


Figure 2: Frequency of submissions per task (blue = immediate feedback support, yellow = no automatic feedback).

The architecture allowing the system to provide immediate feedback consists of two parts: an offline generation process of hypotheses modelling possible well-formed and ill-formed learner answers, and an online matching process that takes the generated hypotheses and matches them in a flexible manner with learner data.

5.1 Offline Hypothesis Generation

The automatic hypothesis generation mechanism works in three steps: i) linguistically analyzing the target answer of an exercise, ii) applying rules to generate alternative forms, and iii) storing the generated forms together with an error diagnosis. In the following, these steps are explained in detail.

As a first step, each target answer of an exercise is analyzed with the help of different NLP tools in order to build a rich linguistic representation as a basis for all further analyses. Table 1 shows the tools employed for analysis.

task	tool
segmentation	ClearNLP (Choi and Palmer, 2012)
part-of-speech tagging	ClearNLP
dependency parsing	ClearNLP
lemmatization	Morpha (Minnen et al., 2001)
morphological analysis	Sfst (Schmid, 2005)

Table 1: NLP tasks and tools

The analyses are encoded in a UIMA Common Analysis Structure (CAS, Götz and Suhre, 2004). A CAS is a source text with multiple layers of annotations, such as a token annotation layer or a dependency-tree annotation layer. By using a DKPro wrapper (de Castilho and Gurevych, 2014) around the UIMA annotators, we ensure flexibility and interchangeability of the specific implementations of the NLP tools.

On the CAS representation of the analyses, we run 40 custom UIMA annotators to explicitly annotate further linguistic properties such as complex tenses or irregular comparative forms. The annotators and the subsequently applied rules described below are designed to cover all grammar topics in the 7th grade English curriculum.

The CAS is then used as input to rules that introduce changes modeling the space of well-formed and ill-formed variability. Some rules introduce changes that yield grammatical forms that are not appropriate in this task context, for example changing the tense of verbs. Other rules generate forms that are never grammatical in any context, such as a regular past tense inflection applied to the lemma of an irregular verb.

When introducing a change, the current CAS is first cloned to yield a deep copy. Then this clone is edited by changing the source text and all linguistic analysis layers that refer to the source text. Furthermore a diagnosis denoting both the type and span of the change introduced as well as the category of the original form is added. The diagnosis thus makes it possible to see what change has been introduced related to which part of the data. If a previous diagnosis was present, it is put into a history list and replaced by the new diagnosis.

For rules generating well-formed alternatives, such as tense changes or contraction expansions, we run the NLP tools used for analyzing the initial CAS on the modified clone and then keep the annotations inside the span that has changed in the rule application. For ill-formed alternatives, we manually encode the linguistic analyses of the changed forms. In any case, the result is a minimally modified clone with an updated, full linguistic analysis. This input-output symmetry makes it possible to apply rules to the output of other rules. This is necessary when chains of rules need to be applied, such as first changing the tense and then altering the verbal morphology of this tense's realization. Each rule is self-contained in that it encodes the conditions under which it applies and the complete logic of the changes when applied.

For the purpose of yielding only desired chains of rule applications and to avoid cycles where two or more rules would add and remove the same forms repeatedly, we group rules in so-called "rule layers". A rule layer is a sorted set of rules that are applied in parallel and do not influence each other. Each of the rules in a layer that is applicable yields a minimally modified clone that serves as input to the second layer of rules. By introducing a "self-copy rule" in each layer we ensure that the original, unmodified target answer percolates through all layers and each rule in a deeper layer can be applied to the original answer as well as to the modified clones.

The algorithm is inspired by graph search algorithms, especially breadth-first graph search (Moore, 1959). In our case, the nodes in the network are CAS data structures with a rule application history, and the edges in the graph are instances of rule applications. An edge can only be traversed if the conditions of applicability defined in the corresponding rule are met. We thus restrict the search space based on task information, here: the linguistic analysis of the target answer(s). The depth of the search tree corresponds to our rule layers. Figure 3 illustrates the process of generating target hypotheses from a target answer by combining multiple layers of rule applications. Table 2 shows a small excerpt from the set of answers generated for a tense and and for a comparative target answer. The table illustrates that the output of any previous layer serves as input to deeper layers. Every hypothesis generated at any layer is saved to the data base.

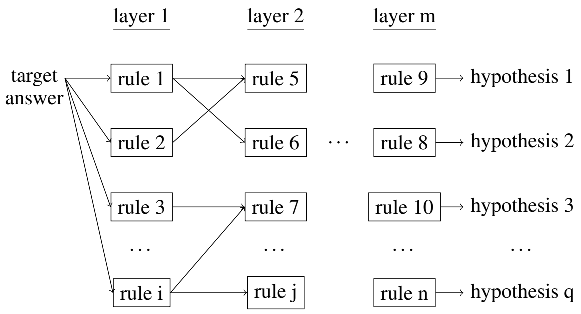


Figure 3: Multi-layered hypotheses generation process

target	layer 1	layer 2	layer 3
are you doing	are you doing	are you doing	are you doing
	were you doing	were you do	was you do
	have you been doing	have you been do	have you been dos
	had you been doing	had you been do	had you been dos
	will you do	are you do	will you dos
	did you do	…	did you dos
	…		are you dos
			was you doing
			is you dos
			is you doing
			…
friendlier	friendlier	friendlier	friendlier
	more friendly	more friendlier	most friendlier
	friendlyer	more friendlyer	most friendlyer
	…		friendliest
		…	friendlyest
			…

Table 2: Examples for generated answer hypotheses

5.2 From Diagnoses to Feedback Messages

To connect error diagnoses with concrete feedback, a language teacher inspected the data we had collected during one year of system use in schools and compiled a list of most common error types made by students with respect to five areas of grammar topics in the curriculum: tenses, comparatives, gerunds, relative clauses, reflexive pronouns. The teacher then formulated error templates for these error types, which specify precisely what linguistic information needs to be present and the (parameterized) feedback message to be generated. To ensure that the conditions under which a teacher would provide a particular feedback and the formulation of the feedback is as close as possible to the real-life educational settings in schools, our project team includes teachers with experience teaching 7th grade English in German high schools, who reduced their teaching load to take on this research project.

Figure 4 shows an example template listing the

Target form:	SIMPLE PAST
Diagnosed form:	SIMPLE PRESENT
Side conditions:	IF-CLAUSE
Feedback message:	"With conditional clauses (type 2), we use the simple past in the if-clause, not the simple present."

Figure 4: Example error template

required target and diagnosed forms as well as necessary side condition along with the resulting feedback message.

Every error diagnosis generated by the system as described above is associated with the most specific compatible feedback template prior to saving a diagnosis in the data base. The system extracts the diagnosis associated with the CAS and all its side conditions, as, for example, signal words for tense forms. For certain phenomena, such as tense confusions, multiple templates exist with varying degrees of specificity depending on the presence of additional linguistic evidence, so that the template providing the best match with the diagnosis can be selected.

The resulting feedback provided by the system for a typical tense error is illustrated in Figure 5. The learner input *will feel* is not correct with respect to the task context requiring present tense. The will future form *will feel* was generated as one of the target hypothesis for the correct target answer *feel*. The student answer in Figure 5 can thus be matched against this generated target hypothesis and the error template associated with this form is displayed as immediate feedback.

5.3 Flexible Online Matching

The generate-and-retrieve approach described above works well for relatively constrained learner input, as it occurs for example with fill-in-the-blanks tasks. However, there are also more open form-oriented tasks in the workbook, where learners have to enter full sentences to practice certain forms, but the lexical material is constrained by the task instruction. In these tasks, students often use slight variations of our pre-computed hypotheses, but make the same systematic errors. Consider the minimal example of an agreement error, as illustrated by the generated hypothesis *he walk*, into which the learner has inserted an additional adverb in *he always walk*. We tackle this issue by allowing for partial matches of target hypotheses, where the obligatory part of the hypothesis must be matched, but an optional remainder can be varied. In the example, both *he* and *walk* would be obligatory to match, whereas *always* is optional.

Technically, the approach is realized via information retrieval on stored target hypothesis forms. We use Lucene (https://lucene.apache.org) for indexing and retrieval, employing the same linguistic pre-processing as in the hypothesis generation step in order to ensure comparability of student answers and target hypotheses. Given a list of hits returned by Lucene, we compare the student input to each of the hits and use the first hypothesis where the student answer satisfies all of the matching constraints.

Figure 6 shows an example from a task where students need to enter the correct tenses in conditional clauses. In the example input shown, the student left out the word *more* that is part of the correct answer, and also used pronouns instead of proper names. But since this is not relevant for the diagnosis of the first tense error here, we can still show feedback based on the stored generated hypothesis. Note that the second tense error, simple present *feels* instead of *would feel*, is handled by a subsequent feedback message once the student submits the update answer. This is in line with previous research on the effectiveness of feedback showing that it is preferable to alert the student of one problem at a time (cf., e.g., Heift 2003).

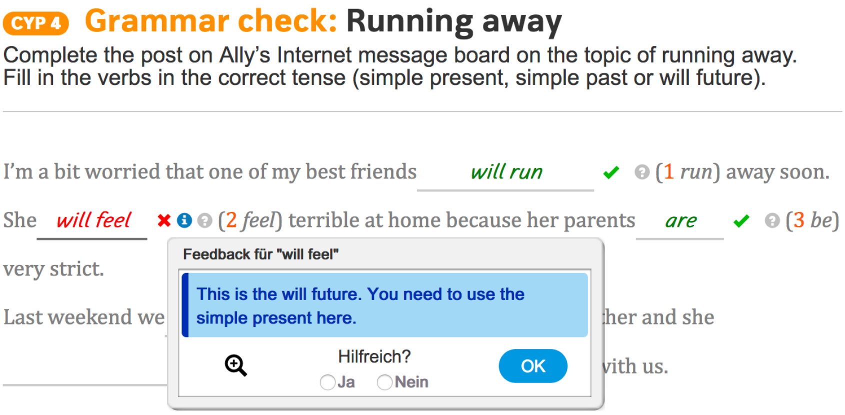
5.4 Individual Immediate Feedback

When students enter an answer into a field of a feedback-enabled exercise, our system executes the algorithm in Figure 7. Using a multi-fallback strategy, the algorithm ensures that more complex feedback retrieval is only tried when simpler strategies (such as a direct match) have failed. Since the student is expected to change their answer upon receiving system feedback, the approach aims at efficiently guiding the student to the correct answer in multiple interactive steps.

6 Evaluation

In this section, we describe an evaluation of the feedback currently given by our system. In a real end-to-end evaluation of a tutoring system, the most interesting evaluation would be to assess the learning gains for the students. We are currently designing a randomized controlled field study for just such an evaluation involving several classes in the coming school year. At this point, however, we can at least report offline evaluation metrics calculated on the student answer data that we collected so far. We plan to make a more comprehensive data set available for research after having conducted the full-year intervention study.

Based on the elicited data introduced in section 4, we selected all individual student answers from the interaction logs of tasks with active, im-



CYP 4 Grammar check: Running away

Complete the post on Ally's Internet message board on the topic of running away.
Fill in the verbs in the correct tense (simple present, simple past or will future).

I'm a bit worried that one of my best friends *will run* ✓ (1 *run*) away soon.

She *will feel* ✗ (2 *feel*) terrible at home because her parents *are* ✓ (3 *be*)

very strict.

Last weekend we … her and she

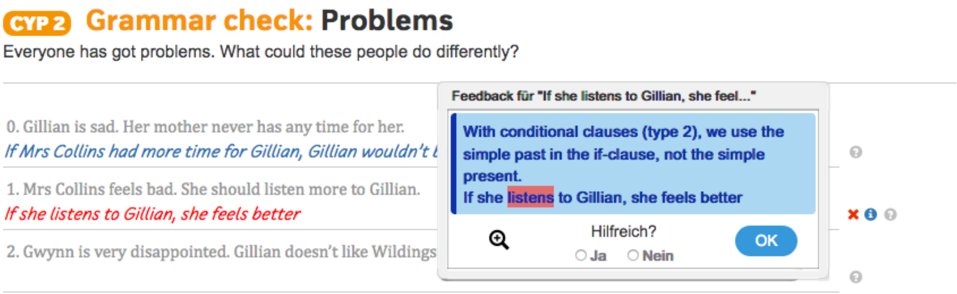
… with us.

Feedback für "will feel"

This is the will future. You need to use the simple present here.

Hilfreich? ○ Ja ○ Nein OK

Figure 5: Feedback on tense error



CYP 2 Grammar check: Problems

Everyone has got problems. What could these people do differently?

0. Gillian is sad. Her mother never has any time for her.
If Mrs Collins had more time for Gillian, Gillian wouldn't …

1. Mrs Collins feels bad. She should listen more to Gillian.
If she listens to Gillian, she feels better ✗

2. Gwynn is very disappointed. Gillian doesn't like Wildings …

Feedback für "If she listens to Gillian, she feel..."

With conditional clauses (type 2), we use the simple past in the if-clause, not the simple present.
If she listens to Gillian, she feels better

Hilfreich? ○ Ja ○ Nein OK

Target answer (for reference):
If Mrs Collins listened more to Gillian, she would not feel so bad.

Figure 6: Student answer including multiple errors with feedback based on a partial hypothesis match

```
if student input == target answer:
  visualize this with green check mark
  -> DONE
else:
  retrieve direct hypothesis matches
  if there are direct matches:
    show associated feedback
  else:
    perform token-level Lucene query
    if there are Lucene hits:
      for every hypothesis:
        if student answer matches criteria:
          show associated feedback
    else:
      show default feedback
```

Figure 7: Feedback algorithm (simplified pseudo-code)

mediate feedback. However, since some of these tasks have meaning-oriented goals (e.g., comprehension, translation), which we do not yet provide feedback on, we excluded data from tasks where the title clearly indicated such a goal (e.g., "Reading: …"). On the other end of the spectrum, we excluded tasks where students only need to enter single characters as part of words.

The remaining set of 33,589 individual student answers (6,755 distinct types) was provided as input to the feedback algorithm of Figure 7.

Note that this data set consists of the authentic learner answers entered into the system at any stage of development. So we run the current version of the feedback algorithm on all the authentic learner data to obtain a complete, current picture of current system performance.

19,809 of the answers were identified as identical to the target answer after basic normalization (upper/lower case, spaces, Unicode punctuation).

Since we do not have gold standard feedback labels for the overall data set, and obtaining them would be a time-consuming annotation task by itself, every student answer that diverges from the target answer must be treated as potentially erroneous and in need of feedback. Note, however, that this diverging set also includes well-formed paraphrases, meaning errors, and form errors we do not intend to provide specific, meta-linguistic feedback on (e.g., spelling).

6.1 Quantitative Results

Table 3 summarizes the results (TA = target answer). We report both answer type counts and answer token counts. For the answers differing from the target answer (i.e., the ones the system provided feedback on), we also report the percentage relative to the total number of answers differing from the target forms.

	# types	# tokens	
identical to TA	342	19,809	
default feedback	5,717	10,297	74.72%
specific feedback	696	3,483	**25.28%**
total	6,755	33,589	

Table 3: Quantitative evaluation results

For the majority of differing answers (74.72%) the system provides default feedback, where a diff with the target answer is shown to the student, as exemplified by Figure 8. As the example illustrates and we will argue in section 6.2, default feedback does not necessarily mean the system missed a potentially relevant error, but can also mean that the default feedback is appropriate or the type of task does not lend itself well to form-focused feedback.

In 25.28% of the differing answers, the system was able to give specific, meta-linguistic feedback, with well-formed and ill-formed tense variation being by far the most productive error pattern. Note that while 696 answer types with specific feedback may seem small, they account for roughly five times as many instances (3,483), showing that it is well worth the effort to model specific, typical error patterns. In comparison, the 10,297 default cases are distributed across 5,717 types, each occurring only about two times, suggesting that there is a long tail of rarely occurring error types that one may not want to model and provide dedicated, meta-linguistic feedback for.

To further analyze this long tail, we calculated the edit distance between the differing answer types and their respective target answers, and investigated the percentage of specific feedback for different edit distance ranges. We found that for the range below the first edit distance tertile, the percentage was at 30.8% and thus higher than the average 25.28%. On the other hand, for the range above the second tertile of edit distances, the percentage of specific feedback is only at 16.6%. The middle range is close to the average, at 25.8%. This suggests that for answers with more variation, including paraphrases and meaning errors, an approach supporting meaning assessment rather than just the form-focused analysis of well-formed and ill-formed variability would be relevant. As a result, we are in the process of integrating the alignment-based CoMiC approach (Meurers, Ziai, Ott, and Bailey, 2011) originally developed for meaning assessment of answers to reading-comprehension questions.

6.2 Qualitative Analysis

Having discussed quantitative results, we now turn to describing several illustrative cases in more detail, using the task displayed in Figure 8.

Example (1) shows a case where the system correctly identifies the systematic problem exhibited by the learner response.

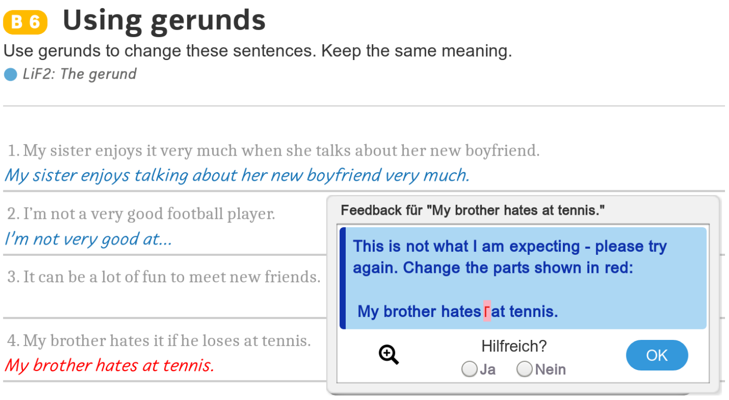
(1) SA: *My brother hates loseing in tennis*
TA: *My brother hates losing at tennis.*
FB: *If an infinitive ends in -e, we leave out this -e with -ing-forms.*

The learner may be unaware of the fact that verbs ending in *-e* drop this suffix in the *-ing* form, and since this is a systematic problem covered by the generation mechanism described in section 5, the system is able to inform the student about this particular challenge to help overcome it. A longitudinal learner model recording typical errors by a user could further support the interpretation and scaffolding of such phenomena.

As an example for default feedback that falls short of pointing out the nature of the learner's error, consider (2) where 'SA' is the student answer, 'TA' is the target answer and 'FB' is the system's feedback. The purpose of the exercise in (2) and the following examples is to practice the use of the gerund, as demonstrated by the target answer.

(2) SA: *My brother's hating it if he lose at tennis*
TA: *My brother hates losing at tennis.*
FB: *This is not what I am expecting – please try again*

Instead of using a gerund ('losing') in connection with the simple present ('hates'), the learner uses an if-clause together with the present progressive ('’s hating'). Additionally, there is an agreement error in the finite verb of the if-clause ('lose' vs. 'loses'). While the general feedback message is not wrong or misleading, a message about the



B 6 Using gerunds

Use gerunds to change these sentences. Keep the same meaning.

LiF2: The gerund

1. My sister enjoys it very much when she talks about her new boyfriend.
My sister enjoys talking about her new boyfriend very much.

2. I'm not a very good football player.
I'm not very good at...

3. It can be a lot of fun to meet new friends.

4. My brother hates it if he loses at tennis.
My brother hates at tennis.

Feedback für "My brother hates at tennis."

This is not what I am expecting - please try again. Change the parts shown in red:

My brother hates at tennis.

Figure 8: Default feedback example

missing gerund or the incorrect verb forms would have been more helpful.

In (3), a learner has provided a different response to the same exercise.

(3) SA: *My brother hates at tennis.*
TA: *My brother hates losing at tennis.*
FB: *This is not what I am expecting – please try again*

Since there is only one error here and it is about the omission of a word ('losing'), the same default feedback that was insufficient in (2) can in fact be helpful enough to guide the student to include a form of the expected word. In a future version, we plan to to include rules targeting the absence of specific grammatical forms, which in this case would enable a more specific message.

7 Conclusion and Outlook

We presented a novel approach to the generation of feedback for English grammar exercises. Building on task properties, we explicitly model the grammar topics targeted by the relevant curriculum (7th grade English) and use a multi-level generation approach to produce the expected range of well-formed and ill-formed variation in student responses to the given tasks. The results of the offline generation process are then used at feedback time in a flexible matching approach in order to account for additional variation in student responses.

Results suggest that the more frequent error patterns are successfully covered by the system, as indicated by the 1:5 ratio of types vs. tokens for which specific feedback is given. In particular, tense-related problems were often diagnosed, which teachers identified as the most challenging grammar topic in the 7th grade curriculum. However, there is also a long tail of infrequent deviations from target answers that do not seem to fall into larger categories. For these, it will be necessary to develop better fallback strategies and evaluate the subjective helpfulness ratings provided by end users at feedback time. Since it is likely that many of the answer deviations occur due to meaning-related issues, our next step will be to integrate meaning error diagnosis into the system. The availability of explicit target answers and the need to diagnose meaning deviations or equivalences between target and student answers suggests that an alignment-based approach such as CoMiC (Meurers et al., 2011) can be effective.

In connection with diagnosing meaning vs. form errors, we also plan to include stronger task modeling into the system. The more we know about the pedagogical goals, the targeted forms, and the range of expected variability, the better we can top-down determine the best feedback strategy before even analyzing a particular student answer.

Finally, we plan to include learner modeling by taking the learners' individual interaction histories into account when providing feedback and for suggesting the next tasks to tackle to provide more practice where needed.

Acknowledgments

We are grateful to our research assistants Madeesh Kannan and Tobias Pütz for their contributions to the implementation of the feedback architecture. We would also like to thank the three anonymous reviewers for their detailed and helpful comments. This work has been funded through a transfer project grant by the Deutsche Forschungsgemeinschaft in connection with the SFB 833.

References

Luis von Ahn. 2013. Duolingo: Learn a language for free while helping to translate the web. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, pages 1–2.

Luiz Amaral and Detmar Meurers. 2011. On using intelligent computer-assisted language learning in real-life foreign language teaching and learning. *ReCALL*, 23(1):4–24.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT)*, pages 1–11, Dublin, Ireland.

Jinho D Choi and Martha Palmer. 2012. Fast and robust part-of-speech tagging using dynamic model selection. In *Proceedings of the 50th Annual Meeting of the ACL*, pages 363–367.

William DeSmedt. 1995. Herr Kommissar: An ICALL conversation simulator for intermediate German. In V. Melissa Holland, Jonathan Kaplan, and Michelle Sams, editors, *Intelligent Language Tutors: Theory Shaping Technology*, pages 153–174. Lawrence Erlbaum Associates Inc., New Jersey.

Thilo Götz and Oliver Suhre. 2004. Design and implementation of the uima common analysis system. *IBM Systems Journal*, 43(3):476–489.

Trude Heift. 2003. Multiple learner errors and meaningful feedback: A challenge for ICALL systems. *CALICO Journal*, 20(3):533–548.

Trude Heift and Mathias Schulze. 2007. *Errors and Intelligence in Computer-Assisted Language Learning: Parsers and Pedagogues*. Routledge.

Ronja Laarmann-Quante. 2016. Automating multi-level annotations of orthographic properties of German words and children's spelling errors. In *Language Teaching, Learning and Technology*, pages 14–22. http://dx.doi.org/10.21437/LTLT.2016-3.

Sébastien L'Haire and Anne Vandeventer Faltin. 2003. Error diagnosis in the FreeText project. *CALICO Journal*, 20(3):481–495.

Alison Mackey. 2006. Feedback, noticing and instructed second language learning. *Applied Linguistics*, 27(3):405–430.

Detmar Meurers. 2012. Natural language processing and language learning. In Carol A. Chapelle, editor, *Encyclopedia of Applied Linguistics*, pages 4193–4205. Wiley, Oxford. http://purl.org/dm/papers/meurers-12.html.

Detmar Meurers. 2015. Learner corpora and natural language processing. In Sylviane Granger, Gaëtanelle Gilquin, and Fanny Meunier, editors, *The Cambridge Handbook of Learner Corpus Research*, pages 537–566. Cambridge University Press. http://purl.org/dm/papers/meurers-15.html.

Detmar Meurers and Markus Dickinson. 2017. Evidence and interpretation in language learning research: Opportunities for collaboration with computational linguistics. *Language Learning*, 67(2). http://dx.doi.org/10.1111/lang.12233.

Detmar Meurers, Kordula De Kuthy, Verena Möller, Florian Nuxoll, Björn Rudzewitz, and Ramon Ziai. 2018. Digitale Differenzierung benötigt Informationen zu Sprache, Aufgabe und Lerner. Zur Generierung von individuellem Feedback in einem interaktiven Arbeitsheft. *FLuL – Fremdsprachen Lehren und Lernen*, 47(2). In press.

Detmar Meurers, Ramon Ziai, Niels Ott, and Stacey Bailey. 2011. Integrating parallel analysis modules to evaluate the meaning of answers to reading comprehension questions. *IJCEELL. Special Issue on Automatic Free-text Evaluation*, 21(4):355–369. http://purl.org/dm/papers/meurers-ziai-ott-bailey-11.html.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–233.

Edward F. Moore. 1959. The shortest path through a maze. In *Proceedings of the International Symposium on the Theory of Switching*, pages 285–292. Harvard University Press.

Noriko Nagata. 2002. BANZAI: An application of natural language processing to web-based language learning. *CALICO Journal*, 19(3):583–599.

Noriko Nagata. 2009. Robo-Sensei's NLP-based error detection and feedback generation. *CALICO Journal*, 26(3):562–579.

Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, and Detmar Meurers. 2017. Developing a web-based workbook for english supporting the interaction of students and teachers. In *Proceedings of the Joint 6th Workshop on NLP for Computer Assisted Language Learning and 2nd Workshop on NLP for Research on Language Acquisition*, pages 36–46. http://aclweb.org/anthology/W17-0305.pdf.

Helmut Schmid. 2005. A programming language for finite state transducers. In *Proceedings of the 5th International Workshop on Finite State Methods in Natural Language Processing*, pages 308–309.

David A. Schneider and Kathleen F. McCoy. 1998. Recognizing syntactic errors in the writing of second language learners. In *Proceedings of the 17th COLING and the 36th Annual meeting of the ACL*, pages 1198–1204, Montreal.

Burr Settles and Brendan Meeder. 2016. A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the ACL*, volume 1, pages 1848–1858.