# Constraints in Free-input Question-Answering Drills

*Lene Antonsen*
University of Tromsø
lene.antonsen@uit.no

ABSTRACT
This article describes a set of question-answer drills for language learning for a richly inflected language. The drills have been in actual use for some time. They allow for free input and make use of a constraint-grammar-based system, which anticipates a number of grammatical errors and common misspellings and gives certain response types. The interactions between student and computer are recorded, and the log reveals that the free-input approach comes at a price: students tend to avoid complex constructions. In order to force the student to answer with more complex constructions, while still keeping the free-input approach, we implemented a solution with more constraints for the input. The exercise items are generated and each template gives rise to a huge numbers of exercises. Constraint grammar makes it easy to control for both grammar errors and adherence to the constraints given in the task. The evaluation on authentic learner data shows that constraining the user's input with the question itself, makes it possible to analyse the student's free input, with very good precision and recall. But parsing the input is only a part of the challenge of designing real-life ICALL systems. The article discusses other design issues related to question-answering drills.

KEYWORDS: Constraint Grammar, ICALL, Grammar Exercises, Syntax.

Lene Antonsen 2013. Constraints in free-input question-answering drills. *Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA 2013.* NEALT Proceedings Series 17 / Linköping Electronic Conference Proceedings 86: 11–26.

## 1 Introduction

Two question-answering drills (QA-drills) offering free input with immediate error feedback, have been available since 2009 for people learning North Saami. This article both presents an evaluation of the processing of the students' input, and how we have met some challenges we have seen from the learners' real-life use of the programs.

The QA-drills are a part of an ICALL system for North Saami called Oahpa! ('Learn!')[1] consisting of both word quizzes and morphological exercises with e.g. fill-in-the-blanks. The system was originally designed as a supplement to ordinary text books. Since 2012 the ICALL-programs have been integrated into the university's introductory courses, and integrated into web-based teaching materials[2], which are used together with teacher instruction.

North Saami is a morphologically complex Uralic language, and its orthographic conventions differ substantially from the native language of most of the students. The language demands a lot of practising before the student reaches the level of fluency required for everyday conversation. Since it is a minority language, with only appr. 18,000 speakers altogether in Norway, Sweden and Finland, learners often do not have enough opportunities to practise the language in a natural setting.

Crucial for learners of North Saami is the mastering of the morphological system, and the QA-drills handled in this paper are based on the concept *Focus on Form*, proposed by (Long, 1991). This means that in the context of a communicative interaction, the attention of language learners is drawn to the form of specific language features. This approach is contrasted with *Focus on FormS*, which is limited solely to the explicit focus on language features, and *Focus on Meaning*, which is limited to focus on meaning with no attention paid to form at all. According to a review on available research (Norris and Ortega, 2000), both Focus on Form and Focus on FormS lead to more substantial effects than implicit instruction.

The main goal of the programs was to develop a language tutoring system with error analysis. Immediate error feedback and meta-linguistic advice about morphology and syntax were seen as important requirements for the programs, grounded in research that shows that formal rule teaching is necessary for adult language learners (DeKeyser, 1995, 2000). Another goal was to make the QA-drills as open as possible for the students, imitating real-word communication with a native speaker.

Even if many ICALL systems of this kind are proposed in the literature, not many of them are fully integrated into real-life foreign-language programmes in universities. In addition to the system presented here, reported systems integrated in university instruction are found for Japanese, Portuguese and German (Nagata, 2002; Heift, 2001, 2010; Amaral and Meurers, 2011).

But being able to process ill-formed input is only part of the challenge of designing real-life ICALL systems; other challenges are how to avoid long instructions in the learners' L1 but still constraining the learner input so that it can be analysed well enough, and how to give appropriate feedback to the learner (Amaral and Meurers, 2011). An evaluation of the program's first three months of operation (Antonsen et al., 2009), and later supervising of the learner data, revealed that the learners' avoidance of complex constructions is a challenge in a free-input system, that the learners' misspellings make the human-computer interaction more

[1] `http://oahpa.no/davvi/`

[2] `http://kursa.oahpa.no/`

difficult and less interesting for them, and that many learners do not work through the whole dialogues. These challenges are the topics of this article, and they are treated in the following sections: The different QA-drills and their design and how we constrain the input is explained in Section 2. All QA-drills use the same analyser, based on finite-state transducers and constraint grammar, which is described in Section 3. Section 4 presents the human-computer interaction, the feedback to the users and an evaluation based on the log files. Section 4.2 looks at some other aspects grounded in direct responses from students. Both the evaluation and how we meet the challenges, are summarised in the conclusion in Section 5.

## 2 The Question-Answering Drills

The drills consist of questions, both of yes/no-questions and wh-questions. The pedagogical goal is to let the student exercise verb inflection by answering the question with correct person, tense and mood, and also use correct case on the noun. The students get tutorial feedback on grammatical errors in their input.

The *Dialogue QA-drill* offers six dialogues built up with ready-made sentences, based on real-world scenarios. In each dialogue there are alternative branches, and the navigation between the branches is made dependent upon the student's answers. For example, if the question is whether the student has a car, a positive answer will navigate to a branch with follow-up questions about the car. Some of the information in the student's input, is stored and used in the questions, e.g. the brand of the student's car or what kind of drink she has chosen. Each dialogue has an underlying pedagogical goal, e.g. the shopping-dialogue is for answering with accusative vs. nominative case, helping a friend moving furniture-dialogue is for answering with locative vs. illative case, and looking at prices in a shop dialogue is for comparison of adjectives.

In the *Open Generated QA-drill* the tasks are made by a sentence generator, in order to be able to create a large number of potential tasks. By tuning the generator, one can easily offer variation to the user, instead of tailoring every task with ready-made questions. The questions come randomly, but are grouped by level of difficulty.

```
    <question>
      <text>Mas SUBJ MAINV</text>
      <element id="SUBJ">
<grammar pos="N"/>
<sem class="FAMILY"/>
      </element>
       <element id="MAINV">
<grammar tag="V+Ind+Person-Number"/>
<id>ballat</id>
      </element>
   </question>
```

Figure 1: Example of a question template. The generated question (in the <text> element) consists of the interrogative *mas* 'what.LOC', the verb *ballat* 'fear.INF' and a noun from the FAMILY set. The generated question can be *Mas vieljat ballet?* 'What were the brothers afraid of?' The sentence generator handles the agreement between subject and main verb.

A template question matrix contains two types of elements: constants and grammatical units for words selected from a pedagogical lexicon of about 2,700 words that are considered relevant for the learners of the language, categorised by semantic sets. The sentence generator handles agreement, such as person and number agreement between the subject and the main verb. The format for the question 'What is/are SUBJ afraid of?' is presented in Figure 2. The noun for the variable SUBJ is drawn randomly from the FAMILY semantic set, which consists of 48 members, and it can be generated as either singular or plural. The agreement with the verbal is handled in the sentence generator. This sentence generator is also used for generating both question and answer for morphology-grammar exercises (Antonsen et al., 2013).

The Open Generated QA-drill and the Dialogue QA-drill give few constraints for the answer. The student is encouraged to answer with a full sentence and with the same verb as in the question, which is a natural way of answering a question in North Saami, but the purpose is also to force the student to inflect the particular verb. The logs reveal, however, that the students will not write more complex language than they have to. Some examples are the following: students will not answer with a complex NP if they can answer with just a pronoun or a noun, and they will not write a time-expression by using a PP, when they can answer with an adverb instead. The price we pay for the free-input strategy is thus that the users are not forced to exercise more complex language skills.

In order to get the users to construct more complex phrases, a new design of the Open Generated QA-drill has been introduced, here called *Constrained QA-drill*. It presents 2-4 lemmas, which should be used to construct the complete answer. This drill type is inspired by *e-tutor*, a program for teaching German to foreigners  (Heift, 2001), in which the possible input is restricted to a set of lemmas which the user must use to construct a sentence. But unlike the *e-tutor* program, the drill in this paper is made by generated tasks, and it uses the same analyser as the other QA-drills, so it allows the student to add more words to the sentence than the given ones.

The lemmas are drawn from semantic sets so there is variation in the exercise items for the students. The system also offers the student the possibility of varying the answer as long as the given lemmas are a part of it. The question in Figure 2 is *Gean deivet gáffádagas?* 'Who did you meet at the cafe?', and for the answer three lemmas are given: *deaivat* 'meet.V', *suohtas* 'funny.Adj', *skibir* 'friend.N'. The QA-pair is glossed in Examples (1) and (2).

(1) Gean deivet gáffádagas?
who.ACC met.PRT.SG2 at-cafe.LOC
'Who did you meet at the cafe?'

(2) Mun deaivat suohtas skibir
I meet.LEMMA funny.LEMMA friend.LEMMA
'I met a funny friend.'

The system accepts many kinds of answers, as long as the three given lemmas form a correctly inflected NP, as presented in examples (3), (4) and (5):

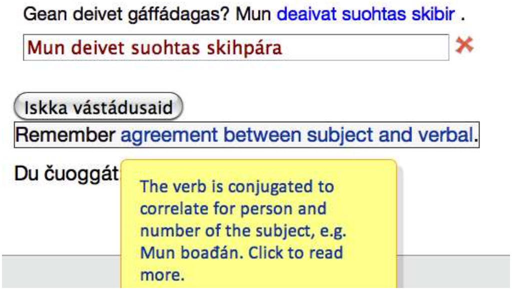
(3) Mun deiven suohtas skihpára.
I met.PRT.SG2 funny.ATTR friend.ACC
'I met a funny friend.'

(4) Mun han deiven iežan suohtas skihpára.
I emph.FOC.PCLE met.PRT.SG2 my.PRON.REFL funny.ATTR friend.ACC
'I (emph) met my funny friend.'

(5) Ikte mun vuot deiven iežan suohtas skihpára.
yesterday I again met.PRT.SG2 my.PRON.REFL funny.ATTR friend.ACC
'Yesterday I again met my funny friend'

The QA-task here is generated from a template with variables:
`<text>Gean MAINV gáffadagas</text> <text>Mun MAINV ADJ NOUN</text>`
The former one is the question and the latter one gives three lemmas for the answer.

The task is thus to inflect the verb and the adjective, which is drawn from a set of 45 suitable adjectives for nouns denoting humans, and the noun from a set of 44 members. Altogether, this template generates 1980 different exercises. Also the verb can be drawn from a set (for this task e.g. *meet, see, know...*), and the number of different exercises expands tremendously.



Gean deivet gáffádagas? Mun deaivat suohtas skibir .

Mun deivet suohtas skihpára

Remember agreement between subject and verbal.

Du čuoggát

The verb is conjugated to correlate for person and number of the subject, e.g. Mun boađán. Click to read more.

Figure 2: A example of a QA-task. The lemmas which the student has to inflect, are marked with blue colour. The feedback in the yellow window is a tool-tip, which appears on the student request, and its has a link to the relevant part of an online grammar. The sentences in the task are explained in Section 2.

Common for all three QA-drills is that the tutorial feedback concerning grammar errors is given in a separate window and the user is allowed to correct the answer until it is accepted. The user can choose the meta language (North Saami, Finnish, Swedish, Norwegian or English) because it is important that they understand the meta-linguistic issues reported by the system. The instructions about how to use the system, very limited, no long explanations, are given in North Saami, but the system offers translations in tool-tip, appearing on the user's request.

## 3 The System and the Analysers

The question-answer pairs are analysed with finite state transducers (FST) for morphology, and a constraint grammar (CG) rule set, which is used both for disambiguating and assigning grammar error tags as triggers for tutorial feedback to the user.

The morphological analyser/generator FST is compiled with the Xerox compilers `twolc` and `lexc` (Beesley and Karttunen, 2003). The lexicon contains 110,000 lemmas – almost half of them are proper nouns. The morphological disambiguator is implemented in the CG-framework (Karlsson et al., 1995).

In order to give better feedback to the students, the FST is enriched with some typical L2 misspellings, which are systematic, and these are marked with error tags (Antonsen, 2012). This makes it possible to some extent to give the student a precise feedback on misspellings, such as ‘X should have consonant gradation’, and offering more explanations about why. In the following example the wordform *addet* (the lemma is *addit*) gets an additional reading as a misspelling of *áddet*, (the lemma is *áddet*). The misspelling is marked with an error tag, `AErr`. This makes it possible, by means of CG-rules, to respond to the misspelling instead of responding to the verb which is actually written:

```
"<addet>" áddet V TV Ind Prs Sg2 AErr  ‘to understand’
"<addet>" addit V TV Ind Prt Sg2       ‘to give’
```

## 3.1 Analysing with Constraint Grammar

For compilation of CG-rules, `vislcg3` is used, this is a new and improved version of the free/open-source compiler `vislcg` (VISL-group, 2008). The program contains manually written, context-dependent rules, mainly used for selecting the correct analysis in case of homonymy. Each rule adds, removes, selects or replaces a tag or a set of grammatical tags in a given sentential context. Context conditions may be linked to any tag or tag set of any word anywhere in the sentence, either locally (in a fixed subdomain of the context) or globally (in the whole context). Context conditions in the same rule may be linked, so that they are conditioned by each other, negated or blocked by interfering words or tags.

The North Saami syntactic analyser based on constraint grammar has an F-score of 0.99 for part-of-speech (PoS) disambiguation, 0.94 for disambiguation of inflection and derivation, and 0.93 for assignment of grammatical functions (syntax) (Antonsen et al., 2010). CG is feasible for grammar checking, and in use for existing grammar checkers, e.g. of Norwegian, Swedish, Danish and Basque (Johannessen et al., 2002; Arppe, 2000; Birn, 2000; Bick, 2006; Uria et al., 2009). There is a prototype for native speakers of North Saami (Wiechetek, 2012). CG is also used in an ICALL program for annotation of free-user input for seven languages (Bick, 2005).

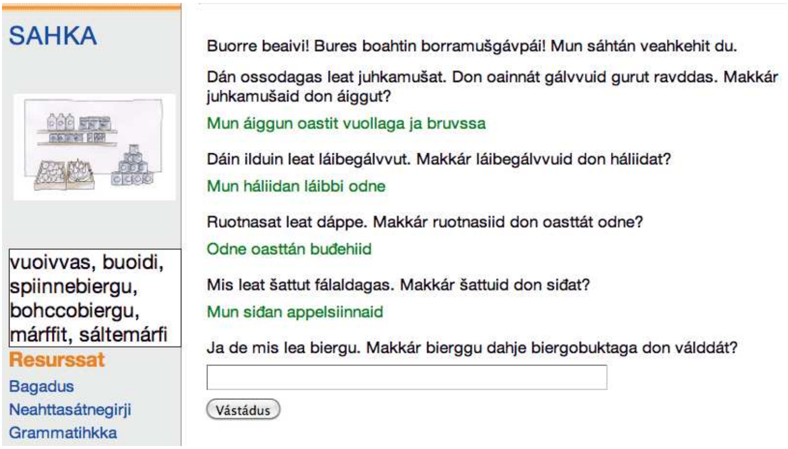
The `vislcg3` rule set used for analysing the input from the drills, consists of two parts. The first part is a set of 872 rules, which disambiguates the user's input only to a certain extent. The rule set is relaxed in comparison to the ordinary disambiguator, in order to be able to detect relevant readings despite grammatical and orthographic errors in the input. The second part of the rule set contains 247 rules for giving feedback to grammatical errors, and for the Dialogue QA-drill there are rules for navigating to the next question or utterance based on the user's answer.

Both the rules for assigning navigation tags and grammar error tags are in the same CG rule set. The advantage of having them in the same file, is saving starting up time, and the flexibility in ordering of the rules. It is e.g. possible to choose dialogue path before commenting on grammar errors. And one can choose to ignore misspellings, which are recognised of the system, in favour to navigate to the next question. To our knowledge, constraint grammar has not been applied for dialogue navigation before.

## 3.2 Feedback on Input

The system gives two kinds of feedback: If the student's answer is accepted, it turns green and the next question is presented, which in the Dialogue QA-drill can be a follow-up question, see

Figure 3. If the answer is not accepted, there will be feedback about the grammatical problem in the answer, and it can even point out the problematic word. Both feedback types are done by CG-rules.



vuoivvas, buoidi, spiinnebiergu, bohccobiergu, márffit, sáltemárfi

Buorre beaivi! Bures boahtin borramušgávpái! Mun sáhtán veahkehit du.

Dán ossodagas leat juhkamušat. Don oainnát gálvvuid gurut ravddas. Makkár juhkamušaid don áiggut?

Mun áiggun oastit vuollaga ja bruvssa

Dáin ilduin leat láibegálvvut. Makkár láibegálvvuid don háliidat?

Mun háliidan láibbi odne

Ruotnasat leat dáppe. Makkár ruotnasiid don oasttát odne?

Odne oasttán buđehiid

Mis leat šattut fálaldagas. Makkár šattuid don siđat?

Mun siđan appelsiinnaid

Ja de mis lea biergu. Makkár bierggu dahje biergobuktaga don válddát?

Figure 3: From the Dialogue QA-drill. The setting is in a grocery and the learner is answering questions about what to buy. The available items show up in the window to the left. The accepted answer turns green, and the next question is presented. The pedagogical goal is to use accusative case in the answers.

The system's question is merged with the student's answer, and given to the analyser as one text string. Instead of a sentence delimiter, the question mark is exchanged with a question delimiter tag (QDL), so that the CG-rules can refer to the question and the answer separately, even if they are merged into one sequence. The question itself constrains the possible analyses of the user input.

The navigation in the Dialogue QA-drill is done by CG-rules, which e.g. assign tags to the string according to whether it contains an affirmative or negative answer, or assign a tag to the target of the question. The rule in Example (6) adds the tag `&dia-target` to the head of the NP, if it is in accusative, and there is no negation to the left of it (*–1), and the interrogative on the left side of the QDL asks for an accusative, defined as the set TARGETQUESTION-ACC. There are exceptions for the possibility of that the targeted word could be a genitive (which is homonymous with accusative) modifying a noun to the right (0 Gen LINK 1 N). The rule is simplified:

(6) ADD (&dia-target) TARGET NP-HEAD + Acc IF (*–1 QDL BARRIER Neg
LINK *–1 TARGETQUESTION-ACC)(NEGATE 0 Gen LINK 1 N) ;

The next question in the dialogue may comment the student's answer by including an inflected form of the `&dia-target`-lemma, or there may be rules navigating to another branch of the dialogue according to the lemma itself, like in the following example. If the student wants tea, she will be offered honey, but sugar if she prefers coffee. The question (the <text> element) is

‘Do you want coffee or tea?’ and the next question will differ according to the tags mapped by CG-rules to the answer. There will always be a default path if the analyser fails to interpret the answer in any of the predefined ways:

```
<text>Háliidat go gáfe vai deaja?</text>
<alt target="coffee" link="sugar_question"/>
<alt target="tea" link="honey_question"/>
<alt target="negative" link="drink_something_else_question"/>
<alt target="default"> link="next_topic"/>
```

The rules searching for grammatical errors in the input are common for all the QA-drills, and may depend on the morphology and the syntax in the question, for example, which case the interrogative asks for, or the verb tense, or the person-number inflection of the verb. Since the students’ sentences are answers to known questions, the error-detection rules can be written accordingly, and problems with long dependencies, often faced by error-detection systems, have not occurred so far. The system also contains rules taking as their scope the right side of the QDL: subject-verbal agreement, NP-internal agreement and the case of nouns and pronouns based on the valency of the verb. The system is conservative and opts for safe error detection rules; false negatives instead of false positives, see Section 4.1.

The error tag in Figure 4 is mapped by a CG rule such as Example (7) (simplified):

(7) MAP (&grm-non-agr-subj-v) TARGET VFIN IF (0 $$PERSON-NUMBER-TAG
LINK –1 (Pers Nom) – $$PERSON-NUMBER-TAG LINK *–1 QDL) ;

This rule maps the error tag to the finite verb (VFIN) if its person-number tag is not the same as for the personal pronoun on the left side. The last constraint, that both the verb and the pronoun are in the answer, is given by asking for a QDL to the left (*–1 QDL).

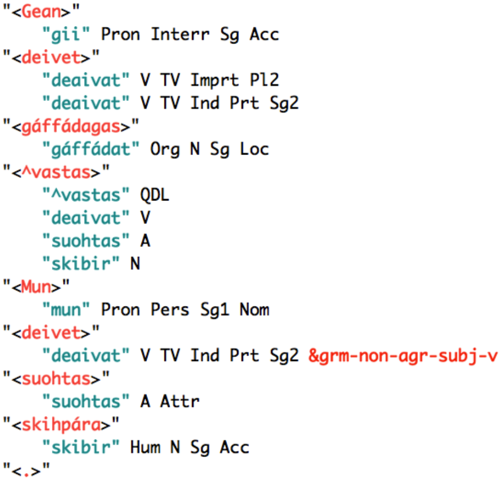
The given lemmas in the Constrained QA-drill are generated from sets in the lexicon and given to the analyser together with the question, stored in the same cohort as the QDL, as in Figure 4. CG-rules map error tags to the string if not all the given lemmas in the QDL cohort are represented in the answer. The handling of the question–answer pair is otherwise the same as for the other drills.

Grammar errors for which there are rules include:

- verbs: finite, infinite, negative form, correct person/tense according to the question
- agreement: subject/verbal, NP-internal
- case of argument and PP based upon the interrogative and valency
- time expressions, some special adverbs, particles according to word order
- comparison of adjectives

The differences between the three types of QA-drills are handled by means of ids assigned by the system to the input, which some of the CG-rules will refer to.

The system gives only one feedback message at a time, even if there are several error tags assigned. The choice of message is decided by the ordering in the message file. The errors based on local context are prioritised, e.g., first giving a message about spelling errors before possible agreement errors, and agreement errors inside the NP are prioritised over agreement between subject and verb, and so on. In some cases two error messages can be triggered by the



```
"<Gean>"
    "gii" Pron Interr Sg Acc
"<deivet>"
    "deaivat" V TV Imprt Pl2
    "deaivat" V TV Ind Prt Sg2
"<gáffádagas>"
    "gáffádat" Org N Sg Loc
"<^vastas>"
    "^vastas" QDL
    "deaivat" V
    "suohtas" A
    "skibir" N
"<Mun>"
    "mun" Pron Pers Sg1 Nom
"<deivet>"
    "deaivat" V TV Ind Prt Sg2 &grm-non-agr-subj-v
"<suohtas>"
    "suohtas" A Attr
"<skihpára>"
    "skibir" Hum N Sg Acc
"<.>"
```

Figure 4: The question and answer pair is given to the analyser as one string. The given lemmas are placed in the same cohort as the QDL (question delimiter) and CG-rules map error tags to the string if they are not represented in the answer. The input is disambiguated to some degree. A CG rule maps the `&grm-non-agr-subj-v` tag to the verb to trigger feedback on the missing subject-verbal agreement. The question-answer pair is translated in Section 2.

same error, and both the error tags will disappear when the student corrects the input after the feedback about the first one.

## 4 Human-Computer Interaction

The design of the QA-drills should match the students' needs, and it is therefore crucial to study the interaction between the user and the system. The recording of the human-computer interaction makes is possible to evaluate and to improve the system.

Immediate error feedback is an important requirement of the system. Above all the feedback should support and facilitate learning, and the error should be seen as a chance of getting the language learner not only to correct the input, but also understand the reason for her misconception.

In Table 1 is an example of a good interaction. The human-computer interaction in Table 2 is not very good. The student gets misleading feedback on the first answer, and has to correct the input a total of three times, before it is accepted.

The users are not homogenous, and it is a challenge to give feedback on the correct level for each learner. By requiring users to log into the system, it would be possible to give individualised feedback. Based on the stored data about their individual performance one could modulate the instructional feedback according to a student model, in which the user is classified as a novice, an expert on so on, on the particular grammar skill. The student model could also be used to provide exercises to the student which focus on their area of weakness. See (Heift and Schulze, 2007) for more.

| *Boađát go odne?* ('Do you come today?') | | |
|---|---|---|
| **user's input** | **System's judge** | **System's feedback** |
| *In sáhtte boahtit odne.*<br>'I don't "can" come today.' | 0 | "sáhtte" has wrong form. It comes after the negation verb and should have negation form. |
| *In sáhte boahtit odne.*<br>'I cannot come today.' | 1 | |

Table 1: An example from the log of a good human-computer interaction. The student corrects her input according to the feedback, and the answer is accepted. System's judge: <1> means that the answer is accepted.

| *Lea go dus heasta?* ('Do you have a horse?') | | |
|---|---|---|
| **user's input** | **System's judge** | **System's feedback** |
| *mus **in** leat heasta*<br>'I don't.Sg1 have a horse' | 0 | Are you confident that you answer in correct person? |
| *mus **in lean** heasta*<br>'I don't.Sg1 have.Sg1=PrtConNeg a horse' | 0 | Are you confident that you answer in correct tense? |
| *m*us **in lea** heasta<br>'I don't.Sg1 have.Sg3 a horse' | 0 | The verb should have negation form. |
| *mus ii leat heasta*<br>'I don't.Sg3 have.Sg3 a horse' | 1 | |

Table 2: An example from the log of a human-computer interaction which is not optimal. The problematic words are here marked with bold. In the first input the negation verb should have agreed with the noun, *ii.Sg3* instead of *in.Sg1*. In stead the student corrects the correct infinite form of the main verb to a form which can be interpreted both as Prs.Sg1 and Prt.ConNeg, and therefore the next feedback comments the tense. A better feedback to the first input would have been: 'Are you confident that "in" is the correct person?'

The system in this paper can be used without logging in, and we have chosen an approach in which the student herself choses how much information she needs. The tutorial feedback is provided to the student on three levels: a short description of the error is always present, on request more information about the grammatical feature is given in a tool-tip, which also contains a link to the relevant part of an online grammar reference (see Figure 2).

Even if the target group for the programs are university students learning North Saami, the programs are freely available on the Internet. One can tell from the usage logs that school pupils use the QA-drills, when they give information about their age and what they do in their answers to the questions in the Dialogue QA-drill. The system's feedback is targeted at students who know the linguistic terminology, and the logs reveal that many users do not always respond to it, probably the young ones, and they do not use the grammar links in the feedback. Therefore they often write many erroneous answers to the same question.

During the first years of operation the dialogues consisted of up to 28 questions in the longest paths through the branches. We learned from the logs that not many students worked through the whole dialogues. We now offer more, but shorter dialogues, each consisting of 8-14 questions, which seem to be a more appropriate number for the students.

## 4.1 Evaluation of the User Log

The evaluation is two-fold, both are based on the real use of the QA-drills. In Table 3 is a comparison of the error feedback the system has added to the users' input, for the same period for the Dialogue QA-drill and Open Generated QA-drill with little constraints for the input, and the Constrained QA-drill.

The misspellings make a large part of the error feedback for all QA-drills, even more often for the Constrained QA-drill than for the other ones. The reason is probably that the user is forced to inflect the given lemmas, and cannot choose to answer with simpler words. Most of the misspellings are systematic, and the FST should have been enriched with more erroneous forms marked with error tags. But even if the users to some extent get specific feedback to the misspellings, the log reveals that the young users don't always understand the meta-linguistic feedback.

For the not-constrained QA-drills 17.7 % of the feedback concerns missing finite verb in the answer. A common reason is that the user answers with a single word. Using the Constrained QA-drill it is clearer for the user that the answer has to contain a verb, and this feedback is quite rare, only 1.3 %.

The feedback on semantics in the Constrained QA-drill concerns using the given lemmas. The evaluation revealed that to Sg2-questions about personal information, the users tended to answer intuitively without the given lemmas. In the Dialogue QA-drill the users sometimes in their answers failed to relate to the objects in the dialogue setting, e.g. to which room they will suggest to put a furniture, even if a list of the available rooms are given in the interface.

| | **Other QA-drills** | | **Constrained QA-drill** | |
|---|---|---|---|---|
| **CG rule type** | **N** | **%** | **N** | **%** |
| misspellings | 307 | 37.5 % | 329 | 43.7 % |
| no finite verb | 145 | 17.7 % | 10 | 1.3 % |
| wrong case/number | 102 | 12.5 % | 133 | 17.7 % |
| verbal-subject agreement | 94 | 11.5 % | 75 | 10.0 % |
| comments on semantics | 89 | 10.9 % | 119 | 15.8 % |
| wrong verb form | 35 | 4.3 % | 36 | 4.8 % |
| NumP internal agreement | 27 | 3.3 % | 22 | 2.9 % |
| other | 12 | 1.5 % | 3 | 0.4 % |
| NP internal agreement | 7 | 0.9 % | 26 | 3.5 % |
| altogether | 818 | 100.1 % | 753 | 100.1 % |

Table 3: Rules in use for a corpus of logged 2834 question-answer pairs.

In the Dialogue QA-drill there are questions about age and family, and a numeral phrase is prohibited in the answer. The feedback about missing agreement inside a numeral phrase makes up for 3.3 % of the feedback, which is almost the same as for the Constrained QA-drill (2.9 %). But the users of the Constrained QA-drill get feedback on NP-internal agreement almost four times as often as the users of the unconstrained QA-drills. The conclusion to Table 3 is that the Constrained QA-drill makes the users form more complex sentences, with finite verb and more complex NPs, than the unconstrained QA-drills.

The other part of the evaluation was to calculate the precision and recall of the system's judgement of the user input. Two parts of the usage log were annotated, for the unconstrained and for the Constrained QA-drill. The results of the annotating are presented in Table 4. Every error-feedback or no-feedback from the system was annotated with true or false.

The precision and recall for the Constrained QA-drill is very good, with the score for precision slightly better than for recall. That means that the system more often slips some errors through, than it flags non-existing errors. For the unconstrained drills precision is not so good: 0.93; the system flags an error which is not there in 7 % of the cases. For 44.7 % of the wrongly flagged errors the reason is a bad CG-rule for accusative in time-expressions. The rule was easy to improve. For some cases there are spelling variants of the word in question, which result in different lemmas in the morphological analysis, and the user gets incorrect feedback on not having used the given lemma or referred to the object in the dialogue. This can be fixed in the FST by unifying variants to one common lemma. In five cases the wordform was missing in the analyser because of limitations done to reduce the compilation time. This proves how important it is to constantly supervise the user logs.

| | True pos. | True pos. but not corr. feedback | False pos. | True neg. | False neg. | Prec. | Rec. |
|---|---|---|---|---|---|---|---|
| Not constrained QA-drills, N=982 | 493 | 44 = 8.9 % | 36 | 439 | 12 | 0.932 | 0.976 |
| Constrained QA-drill, N=1114 | 749 | 72 = 9.6 % | 3 | 352 | 9 | 0.996 | 0.988 |

Table 4: Precision and recall for a part of the user log for the Dialogue QA-drill and the Open Generated QA-drill compared the a part of the log from the Constrained QA-drill.

Important for a good human-computer interaction is that the feedback on the error addresses the error the student has made. For 8.9 % and 9.7 % of the true positives the feedback was misleading, like the example in Table 2. In some other cases the problem was caused by two errors in the same word. The first feedback addressed a specific misspelling. When the student had corrected the misspelling, the next feedback informed the user that the inflection nevertheless was wrong in the context. This illustrates that the algorithm explained in Section 3.2 not always is the best one, and in some cases it has to be modified. Another ordering of the error-messages would have made it possible to comment the inflection despite the misspelling.

Sometimes when the user does not know how to correct the answer, the lemma belongs to a group of stems, which does not follow the general inflectional paradigm. In such cases the feedback should be more specific and address these features for the particular lemma. E.g. even if the main rule is consonant gradation in the stem in certain inflections, there are classes of stems for which this rule does not apply. These words can be recognised by their morphological properties, and it would be possible to give specific comments in the error feedback, e.g. ‘Be aware of that X is a derived agent noun, and therefore it has no consonant gradation.’

## 4.2 Students’ Responses

In some periods there has been a feedback questionnaire on the web. Some users have asked for an answer key to the questions. For the Dialogue QA-drill this is critical because the user is not able to continue the dialogue before the answer is accepted. We have implemented a solution, which allows the user to request for an example-answer after the second time the answer is not accepted. The user still has to type in the answer herself. This was easy to do for the ready-made questions. We will also consider to generate example-answers for the generated questions.

Some students ask for audio files connected to the dialogues. It will be possible to record all the Dialogue QA-drill with ready-made questions. One could first offer only the audio file for the question, and let the student ask for the text, if needed. For the generated questions one could do the same using a text-to-speech program, if the quality were good enough. Such a program for North Saami is under construction now, and will be available in the future.

The evaluation revealed that the young users do not always understand the meta-linguistic feedback. Even if it would have been easier to give a appropriate feedback to the users if by requiring them to log in, we have hesitated to do that, because of the lack of North Saami teaching materials in the whole educational system. A way of giving them a better feedback, but still keep the possibility of using the programs without logging in, could be asking the users to type their age before they start the exercise, and address adapted feedback for the different age groups.

# 5 Conclusion

This article has presented an evaluation of QA-drills based on authentic learner data. One of the drills consists of pre-made questions, the other one uses a question generator, which makes a large number of exercises from each template. All the QA-drills give meta-linguistic feedback to the user, in the dialogue QA-drill a correct answer will be followed up by a further question.

Language learners are generally not able to evaluate the feedback in the way a native speaker does, therefore it is crucial that the system gives appropriate feedback and does not flag false errors. The evaluation of authentic learner data shows that constraining the user’s input with the question itself, makes it possible to analyse the student’s free input with very good precision and recall. For 8.9–9.7 % of the true errors the feedback was still misleading, but for most these instances, it was possible to do better by improving the rules or the ordering of the rules.

The learners’ avoidance of complex constructions is a challenge in a free-input system, but constraining the input with given lemmas to build their answer is a way of getting the learners to write more complex language, while still being within the free-input approach. Constraint grammar is very flexible, and can easily be used to check whether the student really has used the given lemmas in the answer, even when the lemmas are generated from large lemma sets.

Studying the authentic human-computer interaction is important in order to see how the system functions as a whole, e.g. whether the learner carry through the whole dialogue or not, or whether she understands the meta-linguistic feedback, so the input can be corrected. From the logs we see that the dialogue QA-drill often attracts users that do not have enough knowledge of the orthography or the linguistic terminology in the feedback. There is a mismatch between the banal content of the questions in the dialogue, and the necessary skills in orthography and grammar for participating in them. A meaning-based dialogue can loose its meaning when the learner to often is interrupted by comments on grammatical errors.

The constrained QA-drill seems to attract the target group, the students, better, because they give the impression of focusing on lemmas to be inflected and put into correct syntax, instead of trying to give the impression of a real-life conversation.

Despite the lack of a coherent semantic content for the constrained QA drill, the CG-parser gives the computer an intelligent behaviour. This makes it possible to give a sophisticated error analysis and the true interaction between student and computer asked for in (Heift and Schulze, 2007).

## Acknowledgments

Thanks both to my colleague Heli Uibo who has helped me with implementing the QA-drills, and to Norway Opening Universities for the period 2011–13 funding the project Interactive Saami instruction on the Internet, which has been aimed on integrating the ICALL-programs into the university programme. Thanks also to Trond Trosterud for discussions.

## References

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

Antonsen, L. (2012). Improving feedback on L2 misspellings – an FST approach. In *NLP for computer assisted language learning, SLTC 2012*, volume 80 of *Linköping Electronic Conference Proceedings*, Linköping, Sweden.

Antonsen, L., Huhmarniemi, S., and Trosterud, T. (2009). Constraint grammar in dialogue systems. In *Proceedings of the 17th Nordic Conference of Computational Linguistics*, volume 8 of *NEALT Proceeding Series*, pages 13–21, Odense, Denmark.

Antonsen, L., Johnson, R., Trosterud, T., and Uibo, H. (2013). Generating modular grammar exercises with finite-state transducers. In *2nd workshop on NLP for computer-assisted language learning, NoDaLiDa 2013*, volume 85 of *Linköping Electronic Conference Proceedings*, Linköping, Sweden.

Antonsen, L., Trosterud, T., and Wiechetek, L. (2010). Reusing grammatical resources for new languages. In *Proceedings of LREC-2010*, Valetta, Malta. ELRA.

Arppe, A. (2000). Developing a grammar checker for Swedish. In *Proceedings of the 12th Nordic Conference of Computational Linguistics, NoDLiDa 1999*, pages 13–27.

Beesley, K. R. and Karttunen, L. (2003). *Finite State Morphology*. CSLI publications in Computational Linguistics, USA.

Bick, E. (2005). Live use of corpus data and corpus annotation tools in CALL: Some new developments in VISL. In Holmboe, H., editor, *Nordic Language Technology, Årbog for Nordisk Sprogteknologisk Forskningsprogram 2000-2004*, pages 171–185. Museum Tusculanums Forlag, København.

Bick, E. (2006). A constraint grammar based spellchecker for Danish with a special focus on dyslexics. In Suominen, M. e. a., editor, *A Man of Measure – Festschrift in Honour of Fred Karlsson*, volume 19, pages 387–396. The Linguistic Association of Finland, Turku.

Birn, J. (2000). Detecting grammar errors with Lingsoft's Swedish grammar checker. In *Proceedings of the 13th Nordic Conference of Computational Linguistics, NoDLiDa 1999*, pages 28–40.

DeKeyser, R. (1995). Learning second language grammar rules: an experiment with a miniature linguistic system. *Studies in Second Language Acquisition*, 17:379–410.

DeKeyser, R. M. (2000). The robustness of critical period effects in second language acquisition. *Studies in Second Language Acquisition*, 22:499–533.

Heift, T. (2001). Intelligent language tutoring systems for grammar practice. *Zeitschrift fur Interkulturellen Fremdsprachenunterricht*, 6(2).

Heift, T. (2010). Developing an intelligent language tutor. *CALICO Journal*, 27:443–459.

Heift, T. and Schulze, M. (2007). *Errors and Intelligence in Computer-Assisted Language Learning. Parsers and Pedagogues*. Routledge, New York and London.

Johannessen, J. B., Hagen, K., and Lane, P. (2002). The performance of a grammar checker with deviant language input. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1223–1227, Taipei, Taiwan.

Karlsson, F., Voutilainen, A., Heikkilä, J., and Anttila, A. (1995). *Constraint Grammar. A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin and New York.

Long, M. H. (1991). Focus on Form: A design feature in language teaching. In *Foreign Language Research in Cross-cultural Perspective*. John Benjamis publishing company, Amsterdam – Philadelphia.

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

Norris, J. M. and Ortega, L. (2000). Effectiveness of L2 instruction: A research synthesis and quantitative meta-analysis. *Language Learning*, 50(3):417.

Uria, L., Arrieta, B., de Ilarraza, A. D., Maritxalar, M., and Oronoz, M. (2009). Determiner errors in Basque: Analysis and automatic detection. *Procesamiento del Lenguaje Natural*, 43:41–48.

VISL-group (2008). Constraint grammar. `http://beta.visl.sdu.dk/constraint_grammar.html`.

Wiechetek, L. (2012). Constraint grammar based correction of grammatical errors for North Sámi. In *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages (SaLTMil 8 – AfLaT2012)*, pages 35–40.