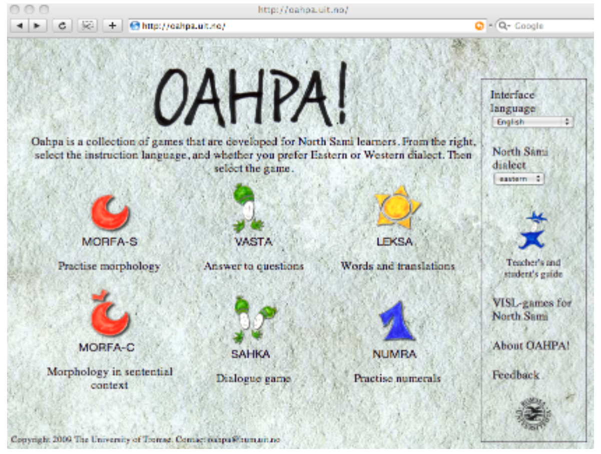
Interactive pedagogical programs based on Constraint Grammar

Lene Antonsen, Saara Huhmarniemi, Trond Trosterud

http://oahpa.uit.no

Abstract

We here present a set of interactive parser-based CALL programs for North Sámi. The programs are based on a finite state morphological analyser and a constraint grammar parser which is used for syntactic analysis and navigating in the dialogues. The analysers provide effective and reliable handling of a wide variety of user input. In addition, relaxation of the grammatical analysis of the user input enables locating grammatical errors and reacting to the errors with appropriate feedback messages.


http://oahpa.uit.no/
OAHPA!
Oahpa is a collection of games that are developed for North Sami learners. From the right, select the instruction language, and whether you prefer Eastern or Western dialect. Then select the game.
MORFA-S
Practise morphology
VASTA
Answer to questions
LEKSA
Words and translations
MORFA-C
Morphology in sentential context
SAHKA
Dialogue game
NUMRA
Practise numerals
Interface language
English
North Sámi dialect
eastern
Teacher's and student's guide
VISL-games for North Sami
About OAHPA!
Feedback
Copyright 2009 The University of Tromsø. Contact: oahpa@hum.uit.no

The Oahpa programs are freely available at *http://oahpa.uit.no*. The programs include a basic morphological exercises (*Morfa-S*), a question-answer (QA) drill (*Vasta*), word quiz (*Leksa*), morphological exercises in a sentential frame (*Morfa-C*), a dialogue program (*Sahka*) and a numeral quiz (*Numra*).

Pedagogical lexicon

The OAHPA! programs share a set of common resources: a pedagogical lexicon and a morphological generator that is used for generating the different word forms that appear in the programs. The dialectal variation is taken into account in the lexicon as well as in the morphology. The lexical entry for *monni* "egg" is given to the right:

```
entry>
  <lemma>monni</lemma>
  <pos class="N"/>
  <translations>
    <tr xml:lang="nob">egg</tr>
    <tr xml:lang="fin">muna</tr>
  </translations>
  <semantics>
    <sem class="FOOD-GROCERY"/>
  </semantics>
  <stem class="bisyllabic" diphthong="no"
      gradation="yes" soggi="i" rime="0"/>
  <dialect class="NOT-KJ"/>
  <sources>
    <book name="d1"/>
    <book name="sara"/>
    <book name="algu"/>
  </sources>
/entry>
```

The morphological properties of words are used when making a detailed feedback on morphological errors. If the user does not inflect the lemma correctly, she can ask for hints about the inflection, and try once more, instead of getting the correct answer straight away.

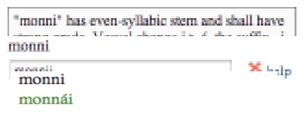
Morphological feedback

The feedback messages are determined by the combination of morphological features in the lexicon and the inflection task at hand. The morphological specification below gives a rule stating that there is a vowel change in illative singular for bisyllabic nouns that end with the vowel *i*. The corresponding feedback message instructs the user to remember the vowel change.

```
<stem class="bisyllabic" soggi="i">
  <msg case="Ill" number="Sg">i_á</msg>
</stem>

<message id="i_á">Vowel change i > á.
</message>
```

The system-internal representation of *monni* states it is a bisyllabic i-stem, which triggers i > á change in illative.



The user types the errouneous *monnii*, and gets feedback from the machine.

A correct answer gets green colour as feedback.

Background and pedagogical motivation

The pedagogical programs in OAHPA! are based upon three pre-existing language technology resources developed at the University of Tromsø: a morphological analyser/generator, a CG parser for North Sámi and a number word generator compiled with the Xerox compiler xfst.

The main goal of the development of OAHPA! was to develop a language tutoring system going beyond simple multiple-choice questions or string matching algorithms, with free-form dialogues and sophisticated error analysis. Immediate error feedback and advice about morphology and grammar were seen as important requirements for the program.

Constraint Grammar (CG)

Constraint grammar is a syntactic framework for choosing correct grammatical analysis of a given wordform, based upon the context it occurs within. Inappropriate analyses are removed, but the last analysis is never removed. CG thus always gives an analysis, and is therefore a very robust framework, well fit to handle potentially erroneous input.

```
"<Makkár>"
    "makkár" Pron Interr Sg Nom
    "makkár" Pron Interr Attr
"<láibegálvvuid>"
    "láibe#gálvu" N Pl Acc
    "láibe#gálvu" N Pl Gen
"<don>"
    "dot" Pron Dem Sg Gen
    "don" Pron Pers Sg2 Nom
    "dot" Pron Dem Sg Acc
"<háliidat>"
    "háliidit" V TV Ind Prs Sg2
    "háliidit" V TV Ind Prs Pl1
"<?>"
    "?" CLB
```

The morphological analyser gives the words in *Makkár láibegálvvuid don háliidat?* "What kind of bread do you want?" all possible grammatical analyses.

```
"<Makkár>"
   "makkár" Pron Interr Attr @>N #1->2
"<láibegálvvuid>"
   "láibe#gálvu" N Pl Acc @OBJ> #2->4
"<don>"
   "don" Pron Pers Sg2 Nom @SUBJ> #3->4
"<háliidat>"
   "háliidit" <mv> V TV Ind Prs Sg2 @FS-STA #4->0
"<?>"
   "?" CLB #5->5
```

The CG grammar then picks the correct analysis, and adds grammatical function and dependency structure.

Evaluation

The overall evaluation shows that the students answer correctly slightly half of the time. By far the most popular program is the basic morphological drill (but the interactve programs have been logged for a couple of days only). The 322 logged **Sahka** errors are distributed along the following lines:

Program	Correct	Wrong	Total	%
Morfa-S	6920	6323	13243	52.3
Leksa	5659	4248	9907	57.1
Numra	3086	2512	5598	55.1
Morfa-C	1349	1613	2962	45.5
Sahka	322	322	644	50.0
Vasta	19	102	121	15.7
Total	17355	15120	32475	53,44

Error type	#	Error type	#
no finite verb	85	wr. case for V-arg	22
orth. error	83	wr. case after Num	10
wrong S-V agr	46	wrong tense	9
no infinite V	30	no postposition	6
wrong V choice	24	wrong word	7

For Sahka we test *precision* (correctly identified errors/all diagnostised errors) *recall* (correctly identified errors/all errors), and *accuracy* (correct judgements/ cases).

Precision = 0.8; Recall = x.y; Accuracy = z.w (N=XXX)

Conclusion

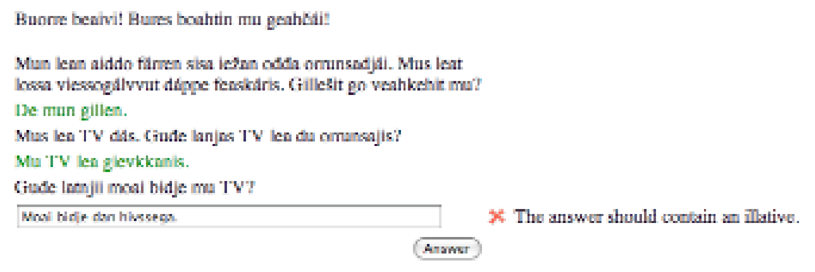
By using the syntactical analyser for North Sámi, combined with a set of error-detection rules, we have been able to build a flexible CALL resource. The programs are modular, and the modules may be improved by adding more materials -- words, tasks, dialogues, levels, words from textbooks. The CG parser framework was originally chosen as parser framework for Sámi due to its extraordinary results for free-text parsing. The present project has shown that CG is well fit for making pedagogical dialogue systems as well.

CG-parser in live analysis in the interactive programs Vasta and Sahka

The programs are based upon free-form interaction: Within certain limits, the student may formulate her own answer.

We use constraint grammar to disambiguate the student's input only to a certain extent, because there will probably be grammatical and orthographic errors. The manually written, context dependent rules are 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. The last part of our grammar consists of rules for giving feedback to the student's grammatical errors, and rules for navigating to the correct next question of in the dialogue, due to the student's answer.

The system question and student answer are analysed together, delimited by the boundary marker ^qst. They first get a morphological analysis (left) , and are then disambiguated, and, if possible, assigned an error tag or a navigation tag.



Buorre beaivi! Bures boahtin mu geahčái!

Mun lean aiddo fárren sisa iežan odda orrunsadjái. Mus leat lossa viessogálvvut dáppe feaskáris. Gillešit go veahkehit mu?

De mun gillen.

Mus lea TV dás. Gude lanjas TV lea du orrunsajis?

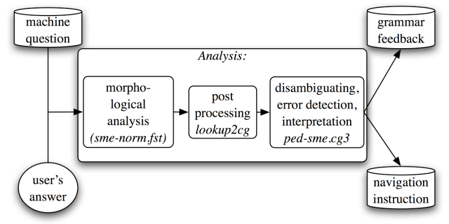
Mu TV lea gievkkanis.

Gude latnjii moai bidje mu TV?

Moai bidje dan hivssegii.

The answer should contain an illative.

Schematical view of the process



$$\begin{tikzcd}
\text{machine question} \arrow[d, dash] & & & \text{grammar feedback} \\
\bullet \arrow[r] \arrow[d, dash] & \text{morpho-logical analysis (sme-norm.fst)} \arrow[r] & \text{post processing lookup2cg} \arrow[r] & \text{disambiguating, error detection, interpretation ped-sme.cg3} \arrow[u] \arrow[d] \\
\text{user's answer} & & & \text{navigation instruction}
\end{tikzcd}$$

Grammar feedback

The system may give feedback, as shown here .....

Here comes an explanation to the example.

&grm-missing-Ill

```
MAP (&grm-missing-Ill) TARGET ("guhte") IF
(1 (N Ill) LINK *1 QDL LINK NOT *1 Ill OR
DOHKO OR Neg BARRIER S-BOUNDARY);

<message id="grm-missing-Ill">The answer
should contain an illative.</message>
```

```
"<Gude>"
   "guhte" Pron Interr Sg Acc
   "guhte" Pron Rel Sg Gen
   "guhte" Pron Rel Sg Acc
   "guhte" Pron Interr Sg Gen
"<latnjii>"
   "latnja" N Sg Ill
"<moai>"
   "mun" Pron Pers Du1 Nom
"<bidje>"
   "bidjat" V TV Ind Prs Du1
   "bidjat" V TV Ind Prt Pl3
"<mu>"
   "mun" Pron Pers Sg1 Gen
   "mun" Pron Pers Sg1 Acc
"<TV>"
   "TV" N ACR Sg Acc
   "TV" N ACR Sg Nom
   "TV" N ACR Sg Gen
"<^qst>"
   "^sahka" QDL gosa_bidjat_TV
"<Moai>"
   "Mo" N Prop Plc Sg Ill
   "mun" Pron Pers Du1 Nom
   "Moa" N Prop Plc Sg Ill
"<bidje>"
   "bidjat" V TV Ind Prs Du1
   "bidjat" V TV Ind Prt Pl3
"<TV>"
   "TV" N ACR Sg Acc
   "TV" N ACR Sg Nom
   "TV" N ACR Sg Gen
"<hivssegis>"
   "hivsset" N Sg Nom PxSg3
   "hivsset" N Sg Loc
   "hivsset" N Sg Gen PxSg3
   "hivsset" N Sg Acc PxSg3
"<.>"
   "." CLB
```

The CG-rules disambiguate and add grammar-error-tag (&grm-missing-Ill) and navigation-tag (&dia-hivsset):

```
"<Gude>"
   "guhte" Pron Interr Sg Acc &grm-missing-Ill
   "guhte" Pron Interr Sg Gen &grm-missing-Ill
"<latnjii>"
   "latnja" N Sg Ill
"<moai>"
   "mun" Pron Pers Du1 Nom
"<bidje>"
   "bidjat" V TV Ind Prs Du1
"<mu>"
   "mun" Pron Pers Sg1 Gen
"<TV>"
   "TV" N ACR Sg Acc
"<^qst>"
   "^sahka" QDL gosa_bidjat_TV &dia-hivsset
"<Moai>"
   "mun" Pron Pers Du1 Nom
"<bidje>"
   "bidjat" V TV Ind Prs Du1
"<TV>"
   "TV" N ACR Sg Gen
"<hivssegis>"
   "hivsset" N Sg Loc
"<.>"
   "." CLB
```

Navigation

Navigating inside the dialogue is implemented in CG-rules. The user input is tagged during analysis with information on whether the answer is interpreted as affirmative or negative. In addition, a special tag indicates whether the sentence contains some information that should be stored for the following questions or utterances. The program is thus able to store simple information such as the student's name, place where she lives and for example the type of her car and use this information in tailored utterances.

In the example to the left the question is "In which room do we put the TV?" One of the alternatives for the navigation is due to the target tag being assigned because of the lemma hivsset ("WC"). The answer will be

```
MAP (&dia-hivsset) TARGET QDL IF (0 (gosa_bidjat_TV))
(*1 ("hivsset") BARRIER ROOMS)  ;
```

There are alternative links in the dialogue, one of them is due to the &dia-hivsset tag:

```
<utt type="question" name="gosa_bidjat_TV">
  <text>Gude latnjii moai bidje mu TV?</text>
  <alt target="hivsset" link="gosa_bidjat_TV">
  <text>Dat gal ii heive! Geahččal oddasit.</text>
        </alt>
  <alt target="default" link="gosa_bidjat_beavddi">
<text>Moai gudde dan ovttas dohko.</text>
</alt>
</utt>
```

Every question has its own unique id, which is used in navigating between questions. In addition, the CG-rules may be tailored for specific questions, like in the rule above.

Age-tags are assigned with help of regex-rules to the answer to the question "How old are you?". The tags function as links for moving to the next dialogue branch tailored to student's age.