Interactive Pedagogical Programs Based on Constraint Grammar

Lene Antonsen, Saara Huhmarniemi, Trond Trosterud
University of Tromsø, Norway, lene.antonsen@uit.no; saara.huhmarniemi@helsinki.fi; trond.trosterud@uit.no
http://oahpa.uit.no

Abstract

OAHPA! is a set of interactive parser-based CALL programs for North Sámi, based on a finite state morphological analyser and a constraint grammar parser which is used for syntactic analysis and navigating in the dialogues. The CG parser provides effective and reliable handling of a wide variety of user input. Relaxation of the grammatical analysis of the user input enables grammatical error detection and reaction to the errors with appropriate feedback messages.

The OAHPA! programs are freely available at *http://oahpa.uit.no*. The programs include basic morphological exercise (*Morfa-S*), a question-answer 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 semantic class is used in the sentence generator for Vasta and Morfa-C. The lexical entry for *monni* "egg" is given to the right.

Morphological feedback

If the user does not inflect the lemma correctly in the morphological exercises, she can ask for hints about the inflection, and try once more, instead of getting the correct answer straight away.

The detailed 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.

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

The user types the erroneous *monnii*, and gets feedback from the machine. A correct answer gets green colour as feedback.

Background and 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! is to make 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 are seen as important requirements for the program.

Due to its complex morphology, Sámi demand a lot of practising before the student reaches a level of fluency required for everyday conversation. Our programs give a practical supplement to the instruction given at school or university. In addition, the dialogue program consists of everyday topics, with underlying pedagogical goals such as practicing verb inflection, choice of correct case form or vocabulary learning.

The sentence generator in Morfa-C and Vasta is able to generate a virtually unlimited number of different tasks, and allows the student to use the programs over and over again.

Constraint Grammar (CG)

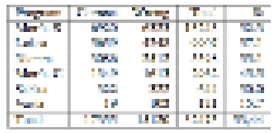
Constraint grammar is a syntactic framework for choosing correct grammatical analysis of a given wordform, based upon the context it occurs within. Each rule removes or selects readings, and adds or removes a syntactic tag. Inappropriate analyses are removed, but the last analysis is never removed. CG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

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

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

Evaluation

OAHPA! has been in use for 3 months, and recieves appr. 500 queries per weekday. 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 Vasta and Sahka have been logged for a couple of days only).



The 322 Sahka errors are distributed along the following lines:

For Sahka we measured *precision* (correctly identified errors/all diagnostised errors), *recall* (correctly identified errors/all errors), and *accuracy* (correct judgements/cases). Precision = 0.8; Recall = 0.68; Accuracy = 0.82 (N=584).

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.

Acknowledgements

Thanks to the faculty of Humanities at the University of Tromsø, and the Sámi Parliament in Norway, for funding the project.

Article version of this poster:

Antonsen, Lene, Saara Huhmarniemi and Trond Trosterud 2009: Interactive pedagogical programs based on constraint grammar. Proceedings of the 17th Nordic Conference of Computational Linguistics. *Nealt Proceedings Series* 4. http://dspace.utlib.ee/dspace/handle/10062/9206

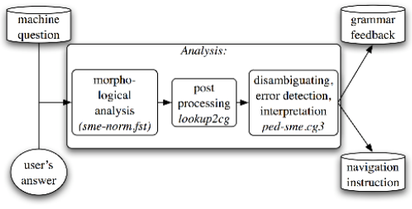
CG-parsing in the interactive free-text analysis programs Vasta and Sahka

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

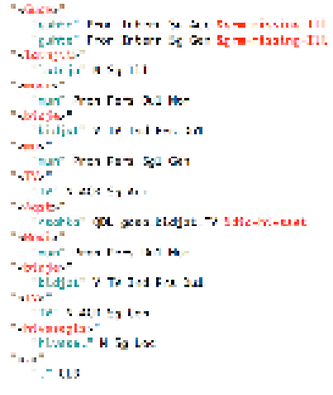
We use constraint grammar to disambiguate the student's input only to a certain extent, because there will probably be grammatical and orthographic errors. The manually written, context dependent rules are mainly used for selecting the correct analysis in case of homonymy. The last part of our grammar 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, depending upon the student's answer.

The system question and student answer are merged and analysed together, delimited by the boundary marker *^qst QDL*. They are first analysed morphologically, and are then disambiguated. If possible, they are assigned an error tag or a navigation tag.

Schematical view of the process



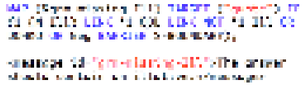
Above is a part of a dialogue in Sahka on furnishing a flat. Below is the analysis of the third question-answer pair from the dialogue. The morphological analysis is disambiguated and a grammar-error-tag (*&grm-missing-lll*) and a navigation-tag (*&dia-hivsset*) are assigned to the analysis:



Grammar feedback

The system may give feedback to grammatical errors. In the third question in the dialogue above, the systems asks "In which room do we put the TV set?" The student answers *Moai bidje TV hivssegis* ("We put the TV set in the WC"), with locative *hivssegis* rather than the correct illative *hivssegii*.

The CG parser disambiguates the input, and the general CG rule below adds a grammar-error-tag (*&grm-missing-lll*) to the sentence analysis triggered by the interrogative pronoun, which demands an illative in the answer.

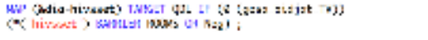


In the grammar feedback library, the tag in question looks up a message in the appropriate interface language (in this example, English), and the user is presented with the feedback *The answer should contain an illative*, as shown in the picture above.

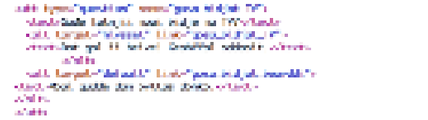
Navigation

Navigating inside the dialogue is implemented by using 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. 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 questions or utterances.

In the example to the left the question is "In which room do we put the TV set?" 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 "That is not a good idea. Make a new try." The CG rule is made for this question-answer pair and assigns the navigation tag (*&dia-hivsset*) to the analysis:



There are several links in the dialogue, one of them is connected to the *&dia-hivsset* tag:



Every question has its own unique id, which is used for navigating between questions. There are both general navigation rules and rules for specific questions, like the one above.

Age-tags are assigned with help of regex-rules to the answer to the question "How old are you?". With help of these tags the system chooses a dialogue branch containing questions relevant to the student's age.