Interactive Pedagogical Programs Based on Constraint Grammar

Lene Antonsen, Saara Huhmarniemi, Trond Trostrud
University of Tromsø, Norway, lene.antonсен@uit.no; saara.huhmarniemi@helsinki.fi; trond.trostrud@uit.no

http://oapla.sit.no

Abstract

OAPAI* 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 tutoring in the dialogues. The CGI parser provides effective and reliable handling of a wide variety of input. Relationalization of the grammatical analysis of the input enables grammatical error detection and reaction to the errors with appropriate feedback messages.

Pedagogical lexicon

The OAPAI programs share a set of common resources: a pedagogical lexicicon and a morphology generator that is used for generating the different word forms that appear in the programs. The lexical set is used in the sentence generator for Vasta and Morti C. The lexical entry for the word “egg” is given in the right.

Morphological feedback

If the user does not infer the lemma correctly in the morphological feedback, the user can ask for hints about the lemma 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 lemma and the inflection task at hand. The correct word form in the lexicicon is a vowel change in illative singular for basic noun roots that end with the stem /j/. The corresponding feedback message instructs the user to remember the vowel change.

Schematical view of the process

The system's internal representation of wrong stems that it is a basic /a/-stem, where the ending /j/ is a change in illative.

The system uses the error as a hint to direct the user to the correct stem. A correct answer gets green colour as feedback.

The questions are based upon free-form interaction: Within the system, the student may formulate her own questions. We use constraint grammar to disambiguate the student's input to a certain extent, because there will probably be grammatical and phonetic errors. The manually written context-dependent rules are mainly used for selecting the correct analysis in cases 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 in the dialogue, depending upon the student's answer.

The programs are based upon free-form interaction: Within the systems, the student may formulate her own questions. We use constraint grammar to disambiguate the student's input to a certain extent, because there will probably be grammatical and phonetic errors. The manually written context-dependent rules are mainly used for selecting the correct analysis in cases 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 in the dialogue, depending upon the student's answer.

Grammar feedback

The system may give feedback in grammatical errors. In the third question in the dialogue above, the system asks "In which month should we place the TV?" The student answers would help TV disambiguating (a grammatical error). The system then gives feedback to the student's grammatical errors and directs him to the correct answer.

Constraint Grammar (CG)

Constraint grammar is a syntactic framework for choosing correct grammatical analyser for 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 CGI grammar then picks the correct analysis, and adds grammatical function and dependency structure.

Evaluation

OAPAI has been in use for 3 months, and receives approximately 500 queries every weekday. 5% of the queries were correct. By far the most popular program is the basic morphological (CG), with usage statistics:

- For Saika and Vasta we measured precise (correctly identified errors/all diagnosed errors), recall (correctly identified erroneous errors), and accuracy (correct judgment/erroneous errors).

- For the error types we target, precision is 0.18, recall is 0.15, and accuracy 0.19 (N=277). But recall than precision indicates that there are very few errors slip through, at the price of erroneously identifying some correct analyses.

- In this pedagogical setting, a goal for future work is improving precision (avoiding erroneous error flagging).

Breaking down the precision numbers on type of feedback, we get the picture below. Of 27 erroneous judgements, 16 were due to technical malfunction, 9 to wrong syntactical 2 to wrong lexical analysis.

Conclusion

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


Navigation

Navigating inside the dialogue is implemented by using CGI 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 then able to add simple static information such as the student's name, place where she lives and for example the type of her car, and use this information in the dialogues.

In the example to the left the question is “In which month should we place the TV?”. One of the alternatives for the navigation is due to the target being assigned being contained by the [nouns] (“nouns”). The answer will be “That is not a good idea. Make a new try.” The CGI rules added for this question-answer pair and assigns the navigation tag ([dilta-av disag]) to the analysis.

There are several links in the dialogue, one of them is connected to the [dilta-av disag] 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 negation-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.