Towards the assessment of free learner's utterances in CALL

Svetla BOYTCHEVA, Irena VITANOVA
Sofia University "St. Kliment Ohridski"
5 J. Bauchier Blvd.
Sofia, Bulgaria, 1164
svetla@fmi.uni-sofia.bg, itv@gmx.co.uk

Albena STRUPCHANSKA, Milena YANKOVA, Galia ANGELOVA
Linguistic Modelling Dept., IPP, BAS
25A Acad. G. Bonchev Str.
Sofia, Bulgaria, 1113
{albena, myankova, galia}@lml.bas.bg

Abstract

Computer-Aided Language Learning (CALL) should play an important role in the modern training process because it provides easy accessible, adaptive and flexible ways of learning. This paper addresses the scenario of tutor-learner question answering and attempts to automate the free answers evaluation using the advantages of Natural Language Processing (NLP). Our current approach integrates shallow parsing for analysing the answers and allows the learners to enter various utterances to express themselves. However this variety does not impede the assessment of the student's answer as we check the utterances against the automatically generated scope of the correct answers. The usage of a "set of answers" instead of one predefined correct answer enables feedback elaboration that helps learners to understand better their knowledge gaps. Briefly, in this paper we show how the combination of shallow and deep semantic NLP techniques can improve the effectiveness of eLearning systems which support communication in free natural language and can make them more satisfactory and pleasant for their users.

1 Introduction and Related Work

Learner-system communication in free Natural Language (NL) is computationally the most challenging and pedagogically the most valuable scenario in CALL. Unfortunately a closer look at the commercial CALL systems shows that most of them support the "free NL input" either by matching user's responses to predefined fixed answers or by manual tutor's checks. The matching is rather superficial, at the level of expected words arranged in expected order. Only few research prototypes try to intelligently cope with the non-trivial task to support the system-student dialogue in (almost free) natural language and to provide some feedback for the user performance.

A frequent target in CALL is to check the syntactic correctness of the learner's utterance. Prototypes like BANZAI (Nagata, 2002), ICICLE (Michaud et all, 2001), CASTLE (Recall, 1997) parse the NL input and provide only feedback for syntax errors.

The first system that makes some steps towards deeper semantic analysis is BRIDGE/MILT (Sams, 1995), (Dorr et all, 1995), see also (Weinberg et all, 1995). The semantic analysis is based on direct matching of the lexical conceptual structures of the user's utterance against the pre-stored lexical conceptual structures of the expected answers. During the matching procedure the system uses synonyms of few concepts and relations. A more sophisticated prototype coping with free NL input is CIRCSIM-Tutor (Glass 2000), which is an intelligent planning-based tutoring system for the domain of cardiovascular physiology. The tutor's questions are closed and the expected answers are quite short. The answers are analysed performing the following steps: lexicon lookup, spelling correction, partial parsing by finite state transducers, lookup in concept ontologies, and finally matching to the question.

A series of intelligent tutoring systems was developed with relation to the Atlas project (Freedman, 1999) which also supports user-tutor communication in free NL. Atlas-Tutor (Freedmanet all, 2000) has a simple NL Understanding module that understands single words, numbers and a few noun phrases and synonyms that are important for the domain. The sentence level understanding of a student's input utterance in Atlas-Andes (Rose et all, 2001) and Why2-Atlas (Van Lehn et all, 2002) is based on the CARMEL Core Understanding Component. The understanding engine of the latter comprises the LCFlex parser (Rose, 2000) and the AUTOSEM (semantic interpretation framework). The parser performs deep syntactic analysis of input utterances. AUTOSEM performs semantic and syntactic interpretation in parallel at parse time in a lexicon driven fashion. To each word in lexical entries syntactic and semantic information is attached. The syntactic information is taken from the COMLEX lexicon. The semantic one is attached by the use of meaning specification representations and rules for mapping between syntactic functional roles and semantic ones. The meaning representations are frame-like structures with slots to be filled in. Knowledge engineers specify the information that the dialogue system must obtain from the user as a set of forms composed of slots. The templates are built using the Carmel-tools authoring system (Rose, 2003). However both the form design and the interpretation as well as the filling algorithm limit the structural complexity of possible dialogues so the approach is not easily scalable. Another complication is that there is a domain specific frame-based language for each domain and thus passing to new domains requires new efforts.

The Geometry Tutor (Alevin et all, 2001) also employs the CARMEL's LCFlex parser in combination with features. The NLU component parses student input using a unification-based approach. The tutor uses the Loom description logic system. The Geometry Tutor classifies the student input with respect to a hierarchy of explanation categories and provides feedback based on this classification.

To conclude, the approaches that semantically process free learners utterances are rule-based and demand extensive knowledge resources. Even when these approaches succeed in the semantic analysis, they failed in its evaluation. The answers are either mapped to pre-stored correct ones or classified into some answer type, which is related to the corresponding feedback type. The feedback is always a static one, i.e. stored in advance. That is why the present prototypes look non-intelligent, non-friendly or toy-like artefacts.

As an attempt to overcome this gap, our paper presents experiments in automatic evaluation of student's free utterances in the financial domain. We apply a combination of shallow parsing and (elements of) deep semantic analysis. The paper discusses in more detail the prototype we developed recently. Section 2 describes our approach including a sketch of the architecture, detailed presentation of mechanism for checking user's answer correctness and discussion of the feedback given to the students. Evaluation results are in summarised in Section 3. Section 4 contains the conclusion.

2 Approaching semantic analysis and relevant feedback

We have developed a self-tuition workbench providing the following activities to the learner: (i) reading teaching materials and (ii) responding to teacher's questions. A pedagogical expert provides in advance the lessons and a number of relevant questions associated to each of them. Please note that the prototype can cope with dynamic question-answering in real dialogue mode despite the fact that at the moment it works over predefined pedagogical resources. The system checks the student's comprehension of financial terms by evaluating answers in free English (in fact all such systems work with controlled English as they deal with a language for special purposes).

For shortness and clarity we describe our approach by examples. Suppose that a user reads a lesson about financial markets, e.g. types of markets, their purposes and kinds of financial instruments traded on them. First the processing module of the workbench parses the lesson.

2.1 Text processor module

GATE (Gate, 2003) performs the lexical analysis and POS tagging. An original left-recursive, top-down depth-first parser in Sicstus Prolog translates the lessons sentences into Logical Forms (LFs). This parser uses grammar rules and rules for translation into LF. Its f-measure is about 91% for the domain we consider. It permits incomplete and some kinds of syntactically incorrect answers and recognises concepts and verbs that are important to the domain. So the parser is somewhat tailored to the financial discourse. A knowledge engineer checks and corrects the LFs if necessary, in order to receive one to one correspondences between the lesson sentences and the generated LFs. So the translation of lessons into LFs is semi-automatic. The predefined questions are traslated to LFs too.

After reading a lesson the student answers to questions, to test his/her comprehension of the material. For instance, let the learner chooses the question:

(1) *What is the function of primary market?*

with corresponding LF:

(1') function(X) & be(Y) & theta(Y, ptnt, X) & theta(Y, obj, Z) & univ(Z) & primary_market(A) & theta (A, poss, X)

The LF (1') used as input for the next module of the system – the scope generator.

2.2 Scope generator

This module automatically generates the correct answer scope using Inductive Logic Programming (ILP) techniques. A specially developed ILP algorithm (Boytcheva, 2002) constructs the scope. The minimum (kernel) and the maximum (cover) are correspondingly the least generalisation and the greatest specialisation under implication of all correct answers. The algorithm processes the set of LFs produced from the lessons (we consider it as a set of all correct answers), the LF of user's answer and some domain knowledge to find minimum and maximum answers. Below we give an excerpt of related statements from the lesson about financial markets concerning the question (1), which help to produce the correct answer scope:

Primary markets are institutional mechanisms set up by society to trade newly issued loans and securities.

The primary market trades new financial instruments and the revenue is used for new investments.

The goal of primary market is to raise capital.

Primary market supports new investments

The usage of a “set of answers” instead of one predefined correct answer makes the system more flexible and capable of evaluating the learner’s answers.

2.3 Answering module

This module analyses the learner's answer and checks its correctness. The shallow parser performs syntactic analysis and logical forms are produced for each sentence. The answering module compares the logical form of the learners’ utterance to the logical forms of the expected minimum and maximum answers, makes the necessary inferences and gives feedback to the learner according to the relative position of the logical forms' terms.

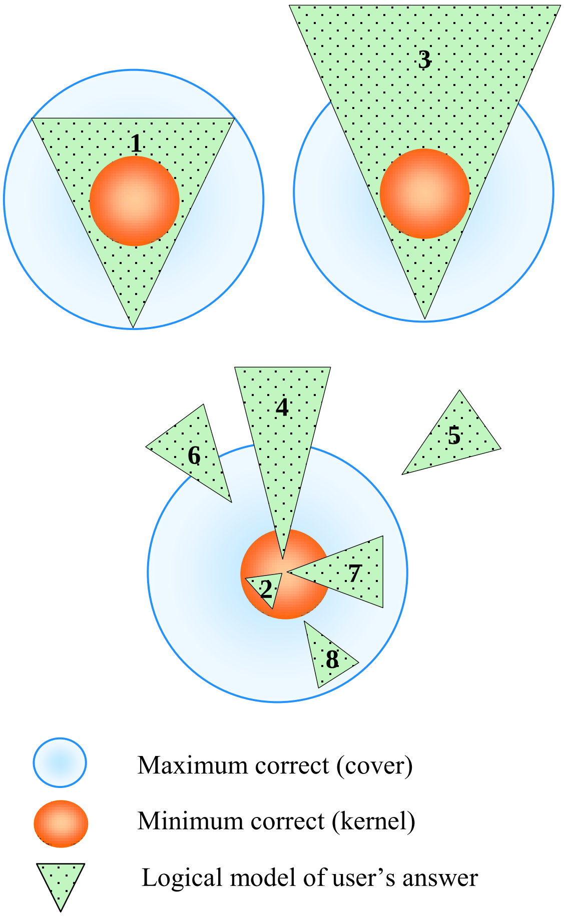

Fig.1 User answer against the correct answer scopes

The possible diagnostics are shown in Fig. 1: : (i) ***correct*** *“configuration 1”*; (ii) ***wrong****“5, 6, 7”*; (iii) ***incomplete****“2, 7”*; (iv) ***more specific*** *“2”*; (v) ***paraphrase****(usage of concept definition instead of the proper term) “2”*; (vi) ***partially correct****“3, 4, 7”*; (vii) ***more general****“3”*.

Figure 1 shows how the module decides about the correctness of the input logical forms. Since there might be many correct answers and their language expression varies considerably, it is not practical to compare the input logical form to a single predefined correct logical form. Rather, the module uses automatically generated scope. The minimum correct answer has to be obligatory included in all the correct answers, i.e. the minimum correct answer is the intersection of all correct answers. The maximum correct answer is the cover of all correct answers. Adding new terms to the maximum answer might be redundant or wrong. There might be several kinds of mistakes in the received answer, so the learner’s utterances have to be investigated with respect to all possible error types.

If the student answers by A1 and A2 to question (1), the diagnostic will be “partially correct, incomplete” for A1 and “partially correct, more general” for A2:

(A1) *The principal function of the primary market is to raise financial capital to support new investment in buildings, equipment and inventories. (7)*

(A2) *Primary market trades financial instruments (4)*

We believe the above-described scenario is the best one for implementation in CALL with free user utterances, moreover it is clear in advance that complete NL understanding of arbitrary sentences is a rather complicated task which cannot be solved in the foreseeable future.

2.4 Knowledge resources

In addition to the pedagogical resources, we use a wide range of domain knowledge: (I) a hierarchy of all important domain concepts, (ii) synonym sets for the concepts and for some of the domain related verbs, and (iii) a knowledge base with assertions and definitions of domain relevant terms as well as definitions of relations. The knowledge resources were developed in a previous project (Angelova et all, 2002).

3 Evaluation

Our previous experience in processing free NL utterances in CALL comes from the LARFLAST project, where we implemented a NLP module that performs deep semantic analysis (Angelova, 2002).

Each student's answer had to contain complete and syntactically correct sentences which was somewhat restrictive for students (adults, non-native speakers with intermediate English proficiency). So we tried to reduce these limitations by applying shallow and partial NLP techniques in Information Extraction style. The students like that they can answer by sentence phrases only in a relatively liberal style and that the word order is irrelevant (as the logical forms are conjunctive terms). At the same time the prototype is not over-permissive as (roughly) each unnecessary word leads to a wrong answer.

We admit however that the semantic analysis as such is a rather expensive task. Defining the domain knowledge and testing the inference and the diagnostics procedures take man-years (in our case this was done in a previous project). Writing the lessons and the related questions in the corresponding way takes time too, exactly as the checks of the logical forms of the lessons and the questions. These efforts make sense only if they are multiplied and the resources are reused in a larger context.

4 Conclusion

Although the task of automatic processing of free NL in CALL is very hard and, similarly to NLU, cannot be solved completely, we believe that the combination of shallow and deep NLP techniques is an attempt to improve the up-to-date CALL solutions especially when the expected learner utterances are relatively short and well-focused. At the same time the communication in NL is more effective and more attractive for the student, so we expect further projects and new attempts to approach the semantic analysis in CALL.

References

A. Weinberg, J. Garman, J. Martin and P. Merlo 1995. A Principle-Based Parser for foreign Language Tutoring in German and Arabic. *In Intelligent Language Tutors: Theory Shaping Technology*, Lawrence Erlbaum Ass. UK, 23-44.

B. Dorr, J. Hendler, S. Blanksteen and B. Migdaloff. 1995. On Beyond Syntax: Use of Lexical Conceptual Structure for Intelligent Tutoring. *In Intelligent Language Tutors: Theory Shaping Technology*, Lawrence Erlbaum Associates, UK, 289-310.

C. Rose, P. Jordan, M. Ringenberg, S. Siler, K. VanLehn and A. Weinstein. 2001. Interactive Conceptual Tutoring in Atlas-Andes. *AI in Education*, J.D. Moore, et al (Eds), IOS Press.

C. Rose. 2000. A Framework for Robust Semantic Interpretation, Proc. of the *1st Annual Conf. of the North American Chapter of the Association for Computational Linguistics (*NAACL'00), 1129-1135.

C. Rose et al. 2003. Overcoming the Knowledge Engineering Bottleneck for Understanding Student Language Input, Proc. of *AI in Ed.*

G. Angelova, S. Boytcheva, O. Kalaydjiev, S. Trausan-Matu, P. Nakov and A. Strupchanska. 2002. Adaptivity in Web-Based CALL, *Proc. of ECAI - 2002*, 445-449.

GATE 2003. See http://gate.ac.uk/

K. VanLehn et al. 2002. The architecture of Why2-Atlas: A coach for qualitative physics essay writing, Proc. *Intelligent Tutoring Systems Conference, volume LNCS 2363*, 158—167.

L. Michaud, K. McCoy and L. Stark. 2001. Modeling the Acquisition of English: an Intelligent CALL Approach, *Proc. of the 8th Int. Conf. on User Modeling*, July 2001, 14-23.

M. Glass. 2000. Processing Language Input in the CIRCSIM-Tutor Intelligent System. *AAAI 2000 Fall Symp. on Building Dialogue Systems for Tutorial Applications*, http://www.csam.iit.edu/~circsim/index.html

M. Sams. 1995. Advanced Technologies for Language Learning: the BRIDGE Project within the ARI Language Tutor Program. *In Intelligent Language Tutors: Theory Shaping Technology*, Lawrence Erlbaum Associates, UK, 17-21.

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

R. Freedman. 1999. Atlas: A Plan Manager for Mixed-Initiative, Multimodal Dialogue. *AAAI-99 Workshop on Mixed-Initiative Intelligence*, Orlando, FL.

R. Freedman et al. 2000. ITS Tools for natural language dialogue: A domain-independent parser and planner, *Proc. Intelligent Tutoring Systems: 5th Int. Conf., ITS 2000, G. Gauthier, et al (Eds)*, Springer: Berlin, 433-442.

RECALL, a Telematics Language Engineering project (1997), see http://iserve1. infj.ulst.ac.uk/~recall. (CASTLE was developed at the IBM Centre in Heidelberg, Germany).

Sv. Boytcheva. 2002. ILP Techniques for Free-text Input Processing, *In Proc. of AIMSA, 10th Int. Conf. on AI: Methodology, Systems and Applications, Springer, LNAI 2443*, 101-110.

V. Aleven, O. Popescu, and K. Koedinger. 2001. Pedagogical content knowledge in a tutorial dialogue system to support self-explanation. *In Papers of the AIED-2001 Workshop on Tutorial Dialogue Systems*, 59-70.