HAL archives-ouvertes.fr

Automated Analysis of Students' Free-text Answers for Computer-Assisted Assessment

Matthieu Hermet, Stan Szpakowicz, Lise Duquette

▶ **To cite this version:**

Matthieu Hermet, Stan Szpakowicz, Lise Duquette. Automated Analysis of Students' Free-text Answers for Computer-Assisted Assessment. The 13th Conference on Natural Language Processing (TALN 2006). April 10-13, 2006. Leuven (Belgium), 2006, Leuven, Belgium. pp.835-845, 2006. <hal-00197397>

HAL Id: hal-00197397

https://telearn.archives-ouvertes.fr/hal-00197397

Submitted on 14 Dec 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Automated Analysis of Students' Free-text Answers for Computer-Assisted Assessment[1]

Matthieu Hermet[1], Stan Szpakowicz[2], Lise Duquette[3]

[1] University of Ottawa, School of Information Technology and Engineering
[2] Polish Academy of Sciences, Institute of Computer Science
{mhermet,szpak}@site.uottawa.ca

[3] University of Ottawa, Second Language Institute
lduquett@uottawa.ca

Résumé

Nous présentons une approche dans le domaine de la Correction Assistée par Ordinateur qui repose sur une analyse symbolique des entrées. La théorie qui soutient cette approche repose sur un travail préalable de mise au point d'un tutoriel d'aide à la comprehension en lecture du français, *Didalect*. Cette théorie doit rendre possible le traitement de segments de texte libre à fins de correction automatique sans l'aide d'une base de réponses pré-encodées. Une étude basée sur un ensemble réduit de réponses étudiantes à plusieurs types de questions nous a permis de justifier notre approche, d'établir une méthodologie et d'élaborer un prototype.

Mots-clés : analyse symbolique, correction assistée par ordinateur, enseignement des langues assisté par ordinateur, tutoriels intelligents, traitement automatique du langage.

Abstract

We present an approach to Computer-Assisted Assessment of free-text material based on symbolic analysis of student input. The theory that underlies this approach arises from previous work on *DidaLect*, a tutorial system for reading comprehension in French as a Second Language. The theory enables processing of a free-text segment for assessment to operate without precoded reference material. A study based on a small collection of student answers to several types of questions has justified our approach, and helped to define a methodology and design a prototype.

Keywords: computer-assisted-assessment, computer-assisted language learning, intelligent tutoring systems, natural-language processing, symbolic analysis.

1. Introduction

The literature on Computer-Assisted Language Learning (CALL) sometimes presents this field in the more general context of Computer-Assisted Learning. This may be fitting, because learning occurs mainly through language. Viewed as a computing and engineering problem, CALL requires contributions in such areas as Virtual Learning Environments (VLE), Computer-Assisted Assessment (CAA) and Intelligent Tutoring Systems (ITS), perhaps controlled by Learning Management Systems (LMS). The success of these complex tools may rely on adaptability and feedback. This calls for solutions motivated by Artificial Intelligence (AI) and Natural Language Processing (NLP).

[1] This work has been partially supported by the Social Sciences and Humanities Research Council of Canada, the program « Initiative on the New Economy ».

CALL is a relatively new field, balanced between the somewhat opposing goals of reusability and didactic specificity. There is a tension between the need to design generic modules (SCORM/ADL, 2006) and very specific systems (for example: Chen and Tokuda, 1999). The former enables the application of sound software engineering principles (without losing sight of commercial potential), while the latter makes didactically attractive solutions possible. Examples of CALL systems with different pedagogical concerns are presented in Graesser (2005), VanLehn (2002), Vitanova (2004) and Michaud (2001). Perez *et al.* (2004, 2005) give an overview of existing CAA solutions and systems as a background in the presentation of their own system.

A prototype VLE *DidaLect* (Desrochers *et al.*, 2004 ; Balcom *et al.*, 2006) has been constructed at the University of Ottawa. It is meant to help students of French as a Second Language (FSL) improve reading comprehesion. It is designed to meet a number of didactic and cognitive goals. Part of the *DidaLect* project is a study of the feasibility of automatic recognition and assessment of free-text answers to certain types of open test questions that the system uses to evaluate the student's progress. This is an autonomous project. Its first prototype will soon be completed, to help assess the validity of the approach. The novelty is that answer assessment does not rely on a comparison with precoded reference material. In this paper we present a theoretical framework of the project and a central procedure. Section 2 briefly discusses *DidaLect*, sections 3 and 4 present our method of assessing free-text answers, and section 5 lists future work and conclusions.

2. DidaLect

DidaLect is an adaptive ITS designed to enhance the reading skills of intermediate and advanced FSL students who work without teacher supervision. The system has a solid grounding in theories coming from the fields of education, cognition and psycholinguistics. Its VLE includes a placement test, a collection of tutorial texts, and resources that assist in the acquisition of reading skills: dictionaries, comprehension tests and *bloc-notes* which help activate prior knowledge. Duquette and Desrochers (2006) present a detailed description of *DidaLect*'s tutoring content.

Text comprehension consists in understanding communication goals expressed in language and accessible through cognitive operations of sense acquisition (understood as sets of *propositions*) and the student's conscious use of such operations (Denhière and Baudet 1992). These, in turn, depend on declarative and procedural knowledge. The former includes syntactic and lexical facts, either encyclopaedic (searchable in a knowledge base) or inferred (by detecting and using the required material in the text). The latter involves mainly iteration and integration: the first as a means of decomposing the steps for structure-building, the second as a means of guaranteeing coherence in the building process. This theory, which underlies *DidaLect*'s implementation, helps delimit the nature of questions that can be asked. We believe that having firm theoretical grounds for the question types is a large part of the job of implementing free-text answer analysis in a CAA module.

Exercises are necessary to assess and help develop comprehension skills indirectly. In language learning, various types of texts are applied, with various types of questions. There are two objectives: variable degree of text and question difficulty, and high adaptability to a student's learning history. *DidaLect* uses informative texts in four categories: cause-effect, problem-solution, comparison and description (Richgels *et al.,* 1987). Questions built upon this textual material range over two categorization scales based on the relation to text structure and cognition, and on the relation to information. The first scale divides questions into Text-Explicit (TE), based on one sentence, Text-Implicit (TI), which require inference between two or more text segments, and Script-Implicit (SI), where inference is to be drawn based on the learner's world knowledge (Wixson, 1983). The second scale categorizes questions as verification, enumeration, comparison, identification, and possibly more. These scales turn the student's interest to the vocabulary and the communication goals on the one hand, and lexical relations on the other. *DidaLect*'s present prototype is built around multiple-choice questions. Our project aims to extend that to some types of free-text answers.

3. Assessment of free-text answers

In CALL systems, assessment of free-text answers is usually understood as a supervised classification problem. The learner's answer is compared to a set of reference answers, matched fully or partially based on syntactic and semantic similarity. We take advantage of the fact that valid answers to topical questions must be extracted from a reference text, which deals with a specific domain. We plan to perform assessment with limited supervision: the only reference answers to be considered are fragments of the text from which the question is created, and reformulations are not precoded as reference material, but left to evaluation through analysis.

The restriction on the forms of valid answers – and therefore on the reformulation possibilities – is guaranteed at the theoretical level by the pedagogical framework in which we conceive the questions. Only TE and TI questions may allow automatic assessment. A TE question comes from a single sentence of the text. We assume that a TI question arises from at most three sentences via an inference restricted to resolving co-reference, so we can still say that all that is needed for an answer is contained in the text. (SI questions require drawing conclusions from world knowledge, which in most cases is absent from the text.) Therefore, we claim that our method of assessing answers can rely entirely on symbolic linguistic analysis.

The didactic emphasis is on second-language acquisition rather than domain lexicons or lexical relations as in Graesser (2005), VanLehn (2002) or Vitanova (2004). This means that all admissible lexical relations are contained in the text fragment used to create the question. As these relations are not part of a domain lexicon, there is no need to precode the semantics of a domain in a knowledge base. A dictionary of synonyms is enough to assess the validity of lexical expressions being reformulated. (This is not the case in closed domains, where polysemy is rare.) Our goal is to consider lexical relations in a reference answer as manifested in surface-syntactic constituents. We do not deal with concepts in any general ontology, that is, we do not work with word meanings recorded in a knowledge base. In the end, lexical relations in a student answer are reformulated using synonyms of the words that participate in these relations.

With this assumption, assessing the correctness of the student's answer consists in verifying the correspondence of the lexical content with the reference answers, with respect to possible role shifting when the answer is reformulated syntactically, such as shifting the patient and agent around the predicate in passivization – an approach already defended in (Rosé *et al.*, 2003). Therefore, leaving aside the question of feedback, the procedure is close to that of Question-Answering techniques, comparing textual fragments for a match. Here, however, any divergence of segments must be either recorded as supplementary content or recognized as a given known type of error.

The resources required include a robust parsing module and a good dictionary of synonyms coupled with a derivational dictionary. In case of parser failure, a list of part-of-speech-tagged words could be a backup resource. We used the Xerox Incremental Parser, licensed to our lab (Ait-Mokhtar *et al.*, 2001; XIP, 2003). It produces several types of output, such as syntactic trees or lists of lexicalized roles and adjuncts, possibly in XML to facilitate post-processing. We also applied the 540,000 word dictionary of synonyms and the derivational dictionary included in *Dictionnaire Intégral* (2006).

4. Processing

Three word lists are produced from the student's input, based on a comparison between the reference (Ref) and the student's answer (S). L1 is a list of words present in S and absent in Ref. L2 contains words absent in S and present in Ref, L3 – words common to both. As the question to be answered is based on a single sentence, or a maximum of three adjacent sentences, reference material comes from a small segment of the text. The idea is to compare the two segments first for lexical and then syntactic similarities. The lists L1, L2, L3 trigger processing based on how they are populated. Typically, empty L1 and L2 with L3 containing all of words present in the segments would signal a correct student's answer, or at least correct words. Often, however, students tend to reformulate sentences, either to express their knowledge or as they feel that the purpose of the exercise is to show reformulation capability. That is why processing usually begins with both L1 and L2 non-empty.

Once these lists have been built, processing continues with the parsing of the sentences Ref and S. Figure 1 shows part of a XIP output for a real student answer, complete with errors. One of the advantages of XIP is its ability to proceed with limited or incomplete knowledge (hence « incremental parsing »), so it produces a parse even in the presence of errors in the input.

```
SUBJ(<approche^approcher:53>,<cardio-vasculaire^cardio-vasculaire:48>)
OBJ(<approche^approcher:53>,<personne^personne:56>)
VMOD_POSIT1(<approche^approcher:53>,<une^un:55>)
NMOD_POSIT1(<cardio-vasculaire^cardio-vasculaire:48>,<rat^rat:51>)
PREPOBJ(<une^un:55>,<à^à:54>)
PREPOBJ(<rat^rat:51>,<d'^de:49>)
DETERM(<rat^rat:51>,<un^un:50>)
```

Figure 1a. XIP output for an erroneous student's answer: « Le cardio-vasculaire d'un rat s'approche à une personne humain – note that the analysis is incomplete

Based on lists L1 and L2, four cases are possible, depending on the degree of reformulation that the student applied:

1. both L1 and L2 are non-empty: reformulation (figure 1);
2. only L1 is empty: S is a minimal expression of Ref (it contains less than what Ref expresses);
3. only L2 is empty: S builds over a maximal expression of Ref (it contains more than what Ref expresses);

4. both L1 and L2 are empty: no reformulation, S is a maximal expression of Ref (it contains no more and no less than what Ref expresses).

```
SUBJ(<est^être:2>,<rat^rat:1>)
OBJ_SPRED(<est^être:2>,<animal^animal:4>)
OBJ(<possède^posséder:6>,<système^système:8>)
COREF_REL(<animal^animal:4>,<qui^qui:5>)
NMOD_POSIT1(<système^système:8>,<cardio-vasculaire^cardio-vasculaire:9>)
NMOD_POSIT1(<système^système:8>,<semblable^semblable:11>)
NMOD_POSIT1(<celui^celui:13>,<humain^humain:16>)
ADJMOD(<semblable^semblable:11>,<celui^celui:13>)
PREPOBJ(<humain^humain:16>,<de^de:14>)
PREPOBJ(<humain^humain:16>,<de^de:14>)
PREPOBJ(<celui^celui:13>,<à^à:12>)
DETERM(<système^système:8>,<un^un:7>)
DETERM(<animal^animal:4>,<un^un:3>)
DETERM_DEF(<rat^rat:1>,<le^le:0>)
CONNECT_REL(<possède^posséder:6>,<qui^qui:5>)
```

Figure 1b. XIP output for a reference sentence: « Le rat est un animal qui possède un système cardio-vasculaire très semblable à celui de l'humain »

Naturally, the above is only valid in what concerns lexical reformulation, with the exception of 4: perfect correspondence of word sets should signal the identity of syntactic structures, though errors in S could occur in agreement and the use of function words. This, however, will trigger different procedures in order to assess the general correctness of S with respect to Ref and general linguistic principles. Cases 2 and 3 call for similar processing, though they are opposite in terms of content sets. In Case 2, the words present in S are a subset of the words in Ref, and conversely in Case 3. We examine them all in turn, in the following combinations:

- Case 1: Synonymy Detection + Building Tree + Syntactic Assessment,
- Cases 2 and 3: Building Tree + Syntactic Assessment,
- Case 4: Syntactic Assessment.

<u>Pre-processing</u>

Prior to parsing, an answer is pre-processed. This involves sentence-level anaphora resolution, part-of-speech (PoS) tagging of the words, and verb tagging. Perez *et al*. (2005) has shown that co-reference resolution is actually a crucial part in processing of language for comparison between a reference fragment and a candidate fragment. For now, anaphora resolution in our system is manual: questions are encoded together with the reference passage in which all co-reference is resolved. While this process is not trivial to automate, we consider it less important in the phase of prototyping other elements of the procedure.

PoS and verb tagging consist in tagging words with (1) PoS information to counter and correct possible parser failures due to tagging errors – a separate list from that of the parser is used in the process – and (2) verb information, namely categories assigned to verbs, which are tagged as action, state (« être ») or attribute (« avoir ») verbs.

Parsing with XIP produces a tree, a set of dependency relations (such as those shown in figure 1) and an XML file containing gender and number information at the word level. After parsing, XIP output is decomposed in order to catch agreement values and therefore check agreement correctness provided via an independent XML file that records gender and number values for each word. The XML format of the file makes this operation easy to implement, and processing agreement can take place independent of (before or after) other steps.

Synonymy Detection

In Case 1, which represents the majority of situations, the procedure begins with searching for synonymy relations between words in L1 and L2. Words from L1 are derived using the derivational dictionary in all categories of words appearing in L2. This is done to detect possible synonymy relations across PoS categories, such as synonymy between the verb « s'approcher de » from S and the adjective « semblable » from Ref in the example in figure 1. An advantage of the synonymy module of *Dictionnaire Intégral* is that, for a given meaning, it shows prepositions which follow verbs. The synonyms, as detected, are recorded and used as anchors to put S and Ref in parallel in the subsequent process of tree building. Any word from L1 that has not been recorded as a synonym is supplementary and therefore signals information not contained in Ref, while any such word from L2 signals information missing from S with respect to Ref.

Tree Building

This step varies depending on the cases. It employs the XIP relations such as those shown in figure 1. In Cases 2 and 3, it amounts to first putting in correspondence the maximal and minimal phrases expressed in S and Ref, starting from the word(s) contained in the non-empty list (L2 in Case 2, L1 in Case 3), up to a match with a word contained in both sentences. Then, complete trees for the whole S and Ref are built, starting from the XIP relations in which the previously detected matches appear. This step enables us to record the missing (Case 2) or supplementary (Case 3) lexemes from S under their syntactic dependency.

In Case 1, the starting points are the synonyms. If several synonyms have been detected, the synonym that appears in most dependency relations is selected to initiate tree building. Recursion in the process takes place on the most promising words. Those words are identical in form and sense. They are, however, far enough in the two different syntactic structures to maximize the possibility of discovering the location of different lexical content in the process of agglomerating the dependency relations that associates these words.

This tree-building operation works on the two structures S and Ref, of which only Ref is known to be correct. The process will naturally halt once all relations inclusive of the synonyms have been extracted and incorporated together with modifiers (_MOD in figure 1). It results in agglomerated relations (<S>) and (<Ref>), such as the temporary relations in figure 2 built from the analysis in figure 1. After agglomerating all relations in which synonyms appear, further lexemes are needed to continue tree building. These come from the list L3 as the words contained in (<S>) or (<Ref>), but not in both, (« rat » and « humain » in the example). This makes the process of tree-building parallel in that (<S>) and (<Ref>) builds over lexical elements of each other.

Eventually, supplementary words from L1 and L2 will appear in the complete trees as modifiers or heads in relations incorporated based on common words or synonyms. We can therefore detect supplementary or missing material within the frame of dependency relations in the agglomerated relations. Controlling the semantic content of S with respect to Ref amounts to comparing their respective constituents, as words, and their bindings. At the same time, role positions occupied by those words may shift when S is a reformulation of Ref. Comparing the constituents amounts to a search for similarity between the two sentences, keeping a record of all variations. Hence the idea of starting with the synonyms, and continuing with common words kept in different syntactic positions due to syntactic reformulation. Besides, we know from experience that reformulated words are usually semantically the most promising: they make the core of a sentence's postulate. Two trees correspond if they contain the same lexical elements, perhaps as synonyms, within equal or equivalent role-predicate structures. There are four categories of such corresponding structures, for constituents centred on a verb, noun, attributive adjective or predicative adjective.

(S) SUBJ(<OBJ(<approcher>,<personne>)>,<NMOD(<cardio-vasculaire>,<rat>)>)
(Ref) OBJ(<posséder>,< NMOD(<NMOD(<système>,<cardio-vasculaire>)>,
<ADJMOD(<semblable>,<NMOD(<*système*>,<humain>)>)>)>)

Figure 2. Agglomerated relations based on synonyms and modifiers.
A solved co-reference appears in italics

Building the trees based on lexical commonality between the two sentences (identity or synonymy) also enables error detection. Stopping the recursive process of building trees signals either the student's mistake or a parsing error. In our example, « humain » does not appear in the XIP analysis of S: « personne » has been encoded in the lexicon as a pronoun. This is a parser error. We have devised a number of heuristics to tackle the issue, shown in the Heuristics subsection.

<u>Syntactic Assessment</u>

This step verifies the correctness of structure S once tree building has been performed for Cases 1-3. The step rests on a set of correct structures as well as typical ill-formed structures. These ill-formed structures come from a collection of student dissertations as hand-written answers to Script-Implicit questions as well as from literature on second-language learning. Questions of this kind, not assessed automatically, call for longer answers containing multiple sentences and thus are a rich source of error material, though our set of such errors is far from complete. Correct structures are understood both in terms of form and equivalence (rules of syntactic reformulation), so they serve a double purpose of verifying the correctness of a syntactic structure and controlling lexical content and lexicon position between two corresponding structures (that of Ref and S). Simplifying, an active sentence structure of the form S[VP[NP1[V]]NP2] is to be read both as a correct structure in itself and as a reformulation of a passive sentence structure of the form S[VP[NP2[V]]PP[P[NP1]]].

The two sets of rules (for wrong and correct structures) have different forms. Intermediate FSL students' typical mistakes tend to be lexical: wrong articles, prepositions and conjugation. Therefore, ill-formed structures work for phrases rather than sentences. They record typical lexical mistakes together with their correction. Correct structures come from various answers. We currently have 29 questions, each with four syntactically different forms of answers. This material might be helpful if chunked into phrases, but insufficient if each sentence is treated as a fixed pattern.

Heuristics

As often in NLP systems, a number of heuristics are used to solve problems arising from errors. Increasing the number of heuristics is an on-going effort. The various heuristics cover the issues that arise from errors introduced by the parser, errors introduced by the student or reformulations at the syntactic and semantico-syntactic level.

Errors introduced by the parser can be corrected when they have lexical reasons (such as « personne » encoded as a pronoun). Therefore, words from Ref and S are pre-processed using a parallel list for PoS tagging with all possible forms for a noun. If an error occurs, the lexicon is automatically augmented with the missing information (XIP offers this possibility), and sentence is sent back for parsing.

Student errors at the lexical level are tackled through heuristics when parsing is impossible. Students tend to mix word forms and use forms inappropriate enough to prevent parsing. This can be addressed partially. We can detect the point of the parsing failure and either send information to the student or derive the word's alternative forms, reshape the sentence for proper parsing and continue processing, keeping a record of the error. The last approach has been tested and is successful in cases where an erroneous word stands as a wrong reformulation of a word present in Ref (a student mistook present participle *semblant* for adjective *semblable*, which halted parsing).

Heuristics for the assessment of syntactic structures actually call for the rules briefly presented for Syntactic Assessment, and are part of that processing step. Verifying semantico-syntactic reformulations is another matter. This corresponds to semantic variation out of the reach of synonymy analysis. Due to the simplicity of the exercises and the reference text, however, we can address this issue at least partially. We have observed it empirically at the verbal expression level (in our example in figure 1, attribution expressed by the verb « posséder » is an instance of this problem: « le rat possède un système cardio-vasculaire » and « le système cardio-vasculaire du rat » have equal senses but different syntactic values). In this respect, lists of current verbs encoded as action, attribute or state verbs have been produced and put to use to facilitate detection of verbs which act in sentence as discourse connector instead of sense conveyer. This is treated as a position problem under the reformulation rules, and a constituent phrase containing one such verb is to be put in relation of equivalence with other forms equal in sense. The latter rules are part of the set of rules used for Syntactic Assessment, so this aspect of verbal expressions is understood as « semantico-syntactic ».

Assessment

Properly speaking, assessment consists of keeping record of all errors in agreement, lexicon, syntax and content. These categories of assessment are of different pedagogical importance and are not treated as equal. Typically, agreement and syntactic judgements have an absolute value, either right or wrong. Lexicon errors are evaluated with respect to a typology of lexical errors, making distinctions ranging from mistake to error, usually from orthography to misconceptions (as having mistaken *semblant* for *semblable*). Finally, the content of S, as minimum in relation to the content of Ref, may or may not yield an error depending on the nature and function of the missing part(s). In our example, having mistaken « cardio-vasculaire » for a noun is both a lexical and syntactic error. Having taken « personne humain » for « celui (*système cardio-vasculaire*) de l'humain » in the enunciation of comparison is a content error leading to a semantic error, not mentioning agreement. Having left out superlative « très » or subordinate « le rat est un animal qui » are not errors.

Error qualification with respect to minimal validity lies in the distinction between phrase modifiers and sentence modifiers (complements or subordinates). Phrase modifiers are evaluated under their PoS categories, which are considered as necessary or superfluous. For example, an adverb is considered less important than an adjective or a noun. Subordinates are evaluated with respect to the nature of the contained verb. Within a sentence, a subordinate with a state or attribute verb is considered less semantically (or informatively) important than one containing an action verb. Sentence complements, such as temporal complements, are considered as superfluous, at least in the present state of our research.

Feedback can vary in size and content depending on the correctness of the student's answer. Several points are salient.

- General validation over Ref to assess if the student's sentence answers the question.
- Syntactic correctness. In case of error, the system should show the correct structure. If, however, most syntactic errors are due to lexical mistakes, the system should show correct lexical forms, explaining *why* an error occurred. Reasons include PoS mistagging, wrong preposition selection or erroneous verb tense.
- Lexicon. Given a typology of lexical errors, feedback gives the category of error plus a correction when possible. This implies, not q uite realistically, that the system *knows* what the student intended to say.
- Range of semantic content with respect to Ref (see the beginning of this section, minimal and maximal coverage).

<u>Support</u>

A fair number of CAA systems give good results, perhaps better than ours, but those systems work under tight knowledge control. For a question, a number of answers have to be encoded, either in the form of referential isolated sentences or as expressed in an annotated corpus. In both cases, the encoding task is substantial. The advantage of our system is that no encoding of reference material is necessary. The question arises as to whether answer fragments in the text should be included in the formulation of a question, or left to a tracing module, in charge of automatically detecting proper segment. This question has been left aside for now, as implementing a tracing module is of little difficulty, especially due to the nature and size of the texts. We nevertheless posed this question as a possibility of providing the means of detecting wrong text segment selection in the student's effort to build an answer (knowing what the student is talking about).

Therefore, all that is needed to feed the system with didactic material is a set of texts and questions, and a slight degree of expertise on the admissible types, TE and TI, when creating the questions.

5. Future Work and Conclusions

We have presented the design of a CAA module for the unsupervised validation of free-text answers to open questions.[2] This module is rooted in the theoretical framework deployed in *DidaLect*, a tool for the improvement of text comprehension. It is an autonomous tool, though it functions as a self-learning system. The present resources are directed to the acquisition of French, but the core of the system is portable from one European language to another if backed with adequate resources. The module is being implemented. The design centred

[2] This is the topic of the first author's PhD thesis in Computer Science, to be completed in early 2007.

around the proper assessment of a small set of 12 TE and TI answers, specifically gathered for this purpose from intermediate-level FSL students.

Future work includes the enlargement of the answer corpus for testing and validation purposes, as well as the inclusion of a word frequency list in order to support the students with difficult and rare words, and a co-reference resolution module. Our main objective for the coming months remains the implementation of the system based on solid software engineering principles.

References

AIT-MOKHTAR S., CHANOD J.-P., ROUX C. (2001). "A Multi-Input Dependency Parser". In *Proceedings of the Seventh IWPT (International Workshop on Parsing Technologies)*. Beijing.

ANCTIL D. (2005). "Characterization of the Notion of Lexical Error in the Framework of Explanatory Combinatorial Lexicology". OLST-RALI, Université de Montréal.

BALCOM P., COPECK T., SZPAKOWICZ S. (2006). "DidaLect: Conception, implantation et évaluation initiale". In *Technologies langagières et apprentissage des langues. Actes du congrès de l'ACFAS* 105. Montréal (to appear).

CHEN L., TOKUDA N. (1999). "A New Diagnostic System for J-E translation ILTS". In *Proceedings Machine Translation Summit*: 608-616.

G. DENHIÈRE G., BAUDET S. (1992). *Lecture, Compréhension de Texte et Sciences Cognitives*. PUF, Paris.

DESROCHERS A., DUQUETTE L., SZPAKOWICZ S. (2004). "Adaptive Courseware for Reading Comprehension in French as a Second Language: The Challenges of Multidisciplinarity in CALL". In *Proceedings of 11th International CALL Conference*. Antwerp: 85-91.

DESROCHERS A., L. DUQUETTE L. (2006). "Appuyer la Compréhension en Lecture à l'Aide d'un Logiciel Adaptatif". In *Technologies langagières et apprentissage des langues: Actes du congrès de l'ACFAS* 105. Montréal (to appear).

Dictionnaire Intégral (2006). http://www.memodata.com.

GRAESSER A.C., PERSON N., LU Z., GEON M.J., B. MCDANIEL B. (2005). "Learning while Holding a Conversation with a Computer".In *Technology-Based Education: Bringing Researchers and Practitioners Together*. Information Age Publishing: 143-167.

MICHAUD L.N., MCCOY K.F., STARK L.A. (2001) "Modeling the Acquisition of English: an Intelligent CALL Approach". In *Proc 8th International Conference on User Modeling*. *Lecture Notes in Artificial Intelligence* 2109. Springer: 14-23.

PEREZ D. (2004). *Automatic Evaluation of User's Short Essays by Using Statistical and Shallow Natural Language Processing Techniques*. Advanced Studies Diploma, EPS, Universidad Autonoma de Madrid.

PEREZ D., POSTOLACHE O., ALFONSECA E., CRISTEA D., RODRIGUEZ P. (2005). "About the Effects of Using Anaphora Resolution in Assessing Free-Text Student Answers". In *Proceedings of RANLP-2005*: 380-386.

RICHGELS D., MCGEE L., LEMAX R., SHEARD C. (1987). "Awareness of Four Text Structures: Effects on Recall of Expository Texts". In *Reading Research Quarterly* XXII: 177-197.

ROSÉ C.P., GAYDOS A., HALL B.S., ROQUE A., VANLEHN K. (2003). "Overcoming the Knowledge Engineering Bottleneck for Understanding Student Language Input". In *Proceedings of AI in Education*.

SCORM/ADL (2006). http://www.adlnet.org.

VANLEHN K., JORDAN P., ROSÉ C.P. (2002). "The Architecture of why2-atlas: a Coach for Qualitative Physics Essay Writing". In *Proceedings of Intelligent Tutoring Systems Conference*.

VITANOVA I. (2004). "Evaluating Integrated NLP in Foreign Language Learning: Technology Meets Pedagogy". In *Proceedings of InSTIL/ICALL symposium*.

WIXSON K.K. (1983). "Postreading Question-Answer Interactions and Children's Learning from Texts". In *Journal of Education Psychology* 5: 413-423.

XIP Parser (2003). *Xerox Research Center Europe Technical Document*.