ALLES: Integrating NLP in ICALL Applications

Paul Schmidt *, Sandrine Garnier *, Mike Sharwood ?, Toni Badia +, Lourdes Díaz +, Martí Quixal +, Ana Ruggia +, Antonio S. Valderrabanos #, Alberto J. Cruz # , Enrique Torrejon !, Celia Rico !, Jorge Jimenez !

* IAI, Saarbrücken, Germany, paul@iai.uni-sb.de
? Heriot-Watt University Edinburgh, UK**,** M.A.Sharwood_Smith@hw.ac.uk
+ Dept. of Translation and Philology, Universitat Pompeu Fabra, Barcelona, Spain, toni.badia@upf.edu
# SchlumbergerSema, Madrid, Spain, Alberto.Cruz@madrid.sema.slb.com
! Departamento de Traducción e Interpretación, Facultad de Comunicación y Humanidades, Universidad Europea de Madrid, Spain, enrique.torrejon@uem.es

Abstract

This paper describes how mature NLP that has been successfully applied in the area of controlled language checking can be used to deliver intelligent CALL applications1. It describe how an autonomous, long-distance second-language learning system for advanced learners[2] can be created. The architecture of the system consists of a multimodal user interface, a set of skill-specific learning tools, and a set of NLP-based evaluation tools. All modules are integrated in a flexible and scalable software architecture allowing for the use of NLP and ensuring a user-friendly environment based on advanced concepts in language didactics. The multimodal user interface is web-based and incorporate off-the-shelf ASR. The set of skill-specific learning tools consists of a reading, listening, speaking and a writing tool. The thrust of the project is to show the potential of NLP in evaluation of students' productions.

Introduction

This paper describes how mature linguistic resources (successfully applied in controlled language checking) are used for CALL applications to deliver an autonomous, long-distance 2nd-language learning course for advanced learners. The architecture of the system consists of a multi modal user interface, a set of skill-specific learning tools, and a set of NLP-based evaluation tools. All the modules are integrated in a flexible and scalable architecture providing a user-friendly environment. The set of learning tools includes reading, listening, speaking and writing tool. The NLP tools incorporate orthography, grammar and style checking facilities, a tool for evaluating linguistic richness and a domain-specific and exercise-centred content checking. The system will be implemented as a client-server architecture that allows for long distance remote access through Internet. The user interface will be based on web services and will be developed in a way to assure a high degree of usability.

With regard to the learning module, there is an elaborate didactic design that is the basis for the specification of the learning sequence and the learning units which are composed of sets of tasks. Syllabuses have been designed in accordance with European and national standards for the teaching of second languages. The compilation of teaching material follows the principle of felicity and authenticity. NLP tools are used to evaluate learners' output in a self-learning environment. They have been applied already successfully in for example controlled language checking. There is spell checking, grammar and style checking for non constrained language, for constrained language content and semantic checking. ALLES NLP tools provide the means with which to evaluate students' production. The evaluation can be done from at least two viewpoints:

1. Linguistic **correctness**
2. Linguistic **richness**

The most important aspect of this distinction is the fact that assessing linguistic correctness implies that the input text may contain ill-formed sentences, whereas assessing linguistic richness implies that the input text is correct. ALLES will produce tools and strategies that enable the automatic evaluation of written and oral texts both ways.

Assessment of linguistic correctness

As suggested above the task of evaluating linguistic correctness of a text is determined by the kind of errors the system expects from the learner. Traditionally, linguistic checking means correction of orthography, use of grammatical structure, adequate semantic use of words, and discourse structure. However, during the relatively recent history of NLP, only spell-checking and, partially, grammar checking have been tackled (Kukich 1992). This is a consequence of the fact that automatic evaluation of semantics and discourse seems to depend on full natural language understanding (which is not available for unrestricted text). ALLES will, however, explore how NL checking could be used beyond spell and grammar checking to limited contexts and based on pattern matching.
An important consequence of this is that ALLES tools will have to be developed differently for global checking and exercise-specific checking: Global checking is about the detection and diagnosis of errors that may occur in any communicative context. Exercise specific checking refers to the detection and diagnosis of errors that are exclusively applicable in certain communicative contexts.

1 The paper is based on work done in the ALLES project (Advanced Long-distance Language Education System) (IST-2001-34246).
[2] The system is provided for four languages: English, Spanish, German, Catalan and confined to the domain of business and economy.

Global checking techniques

Spell checking: The most common and reasonably tackled task is spell checking. The usual procedure basically consists in (1) detecting all words that are not present in the lexicon, and (2) providing a list of correction candidates by means of applying a minimum edit distance algorithm. A typical problem of this approach is the fact that the number of entries in the lexicon is directly related to the number of non-words found by the system3. ALLES will provide the checking tools for all the languages involved. Experience from other projects shows that even this relatively simple functionality is a great achievement and not at all trivial (e.g. for German, capitalisation and compounding (one word vs. separation) are a nightmare to model.

Grammar checking: Grammar checking has been the most common task in language correction research. Some efforts have been made using statistical techniques. They resulted in either domain-dependent (not very scalable) approaches or were unsuccessful: (St-Onge 1995, Hirst and Budanitsky 2001). ALLES techniques are rule-based. Though numeric techniques have proven to be quick and easy to implement and robust, they appear to be obscure. Our approach reflects human knowledge and is thus easier to grasp. Despite the fact that this has been criticised to be time-consuming the results are at least comparable (if not better). In addition, ALLES grammar checking tools are designed for foreign language learners. This has direct implications for the development of tools, since most of the errors that foreign language learners make are different from those of native speakers.

Restricted NL-checking

Checking semantic or even pragmatic well-formedness seems to create problems as full language understanding on a broad basis seems to be required (deep parsing, inferential interpretation) which is not available. There are two ways out: One is to make sure that the checking only refers to single syntactic or semantic items in an utterance. The other way is to constrain the modelling of understanding to very narrow domains and to a small text corpus. ALLES confines sophisticated NL checking to specific exercises which provides the required constraints and concentrates on the first alternative. There are the two classes of error checking beyond syntax, one is the specific semantic checking of well-formedness, the second the checking of formal correctness of speech acts

Semantic checking:

There is checking of semantic correctness in three ways:

- The appropriate use of words in a certain context
- An appropriate combination of words to build complete and semantically well formed sentence.
- Semantic appropriateness dependent on dialogue state.

The last is not feasible. The first alternative is extremely relevant. Speaking a foreign language the appropriate choice of words is important. Automatic checking of errors would be huge progress for CALL. Heringer[4] gives an extensive account of errors on word level made by learners of German.. E.g. the two German verbs ‘interpretieren’, and ‘dolmetschen’ are both ‘to interpret’ in English. The first reading denotes the concept ‘explain’. In this sense you interpret a sonata, the silence of a partner, or a political situation. The second reading is a specific one, to ‘orally translate’. For learners of German this distinction is a problem. They frequently use ‘interpretieren’ instead of ‘dolmetschen’..

Even more difficult are cases like the following. Three German verbs ‚gehören’, ‚gehören zu’, ‚angehören’ could be translated as ‚belong to’ in English:

Das Haus gehörte dem Mann.
(The house belonged to the man.)
Der Mann gehörte der Universität an.
(The man belonged to the university.).
Der Mann gehörte zu der Universität.
(The man belonged to the university).

However, there are subtle semantic differences as the following (semantically incorrect) examples show:

.
**Der Mann gehörte der Universität.*
(The man was owned by the university.).
**Das Haus gehörte dem Mann an.*
(The house was a member of the man.).

(.). Leaving aside the figurative reading (expressing that the man is ‘eaten up’ by his work at the university, working over time etc.) we have the following facts: ‘gehören’ with dative denotes ownership, ‘gehören zu’ and ‘angehören’ membership.

The ‘gehören / angehören’ case is not trivial. A strategy is to check if there is a wrong combination of main verb and nominal elements. Valency, though, not globally available is available can be provided for a limited vocabulary. Tagging provides lexical semantic information in our system. ‘Haus’ is ‘location/agent’, ‘Mann’ is agent, ‘Universität’ is agent / building. Thus exclude:

**Das Haus gehörte dem Mann an.*

A checking would take into account that ‘angehören’ needs a kind of ‘collective’ as an indirect object. ‘Universität’ though not a ‘collective’, but ‘agent / building’ can fill the slot, such as ‘school’, ‘bank’ etc.. All of them may occur as an indirect object of ‘angehören’.

Speech act correctness

A second area that can be modelled with the same techniques is the (formal) correctness of speech acts. Checking uses syntactic and semantic information. If the task is to reject the following request,

Could you please lend me your car?

possible correct answers are:

No, I won’t.
No, I cannot do that.
I will not lend you my car.

3 Notice that many proper nouns, abbreviations, etc. are absent in most lexicons.

4 See Heringer 2000

Of course, not.
……

It would be possible to check whether negation has been done properly. In addition, it can be determined if the speech act has certain incorrect items. If the utterance contains errors, appropriate messages can be issued.

Strategies and tools

The most important feature of the general strategy and the tools for error checking applied in ALLES is not to start from the concept 'well formed sentence', but detect the type of error directly. It is specifically the expected errors that are encoded. The strategy has been tested in other contexts. This is true for one of the tools applied, KURD (Carl et al. 1997), a formal language to be explained presently. The second tool provides a flexible automatic comparison of students' productions with a set of correct answers.

The KURD formalism takes as input feature structures which are a result of a morphological analysis. The example below represents the German word 'der'.

```
{lu=d_art,c=w,sc=art,fu=def,
        agr={gen=f,nb=sg,case=d;g};{gen=m,nb=sg,case
        =n};{nb=plu, case=g}};
{lu=d_rel,c=w,sc=rel,fu=np,
        agr={case=n,g=m,nb=sg};{case=g;d,nb=sg,g=f}}
```

'der' has three readings as article: feminine, singular, (dative or genitive), masculine, singular, nominative or plural, genitive. 'der' is also a relative pronoun, again with several readings. KURD defines patterns which map onto the morphologically analysed input strings. If the mapping is successful, modifications of the analysis are done according to the specifications in the rule. The formalism has elements of unification systems (KURD: kill, unify, replace, delete). A KURD rule consists of a description and an action part, the description of a number of conditions that must match, successive feature structures representing words. They are marked for modification in the action part. A rule fails if the set of conditions does not match. In this case the action part of the rule is not executed. It is if all conditions apply.

.

An illustration is how German detachable verb prefixes (often ambiguous with prepositions) are disambiguated. So, morphological analysis often has two interpretations. However, the syntactic position of prefixes and prepositions is different. While prepositions occur as the head in prepositional phrases, detached prefixes occur at the end of the sentence followed by punctuation or a coordinator.

```
disambiguate_prefix =
        Ae{c=w,sc=p}e{c=vpref},
        a{c=w,sc=punct;comma}:
        Au{c=vpref}
```

The rule consists of two conditions in the description part and one action in the action part. The first condition matches a preposition ({c=w,sc=p}) and a prefix ({c=vpref}). The word is expected to be ambiguous with respect to its category. Both features c=w and sc=p must co-occur in (at least) one interpretation of the matched word description. The existential quantifier, 'e', preceding the feature structure means that there is an appropriate interpretation in the word description, i.e. there is a non-empty intersection of the feature structure and the word description. The second condition consists of one test. The feature structure matches an end of sentence item {sc=punct;comma}. Here, the universal quantifier 'a' requires the word description to be a subset of the feature structure i.e. there is no interpretation in the word description which is not an end-of-sentence item (see Carl et al. 1997). A word description for which the first condition is true is marked by the marker 'A'. The rule applies if the second condition is also true for the following word description. The action part consists of one action. The word description which has been marked in the description part is unified with the feature structure ({c=vpref}) of the action. This results in an identification of the prefix. The prepositional reading is thus out. How this formalism is used for checking may be explained by another example. A frequent error for learners of German is the placement of commas in front of a coordinator.

```
gram_G463_comma_too_much_coord_Subj=
        a{cat=comma},
        Ae{lu=und;oder},
        a{markcl=ns}
        :Au{gram=gDAF4631de}.
```

This rule says that if there is a comma followed by an 'und' or an 'oder' and items that are marked for 'ns' (subordinate clause), then unify a feature 'gram' with the value 'gDAF4631de' into the feature structure bound by the variable A. The value for the feature 'gram' is a code for an error message that is generated by the system saying 'Wrong placement of comma'.

'K-DIFF'

K-DIFF is a tool that compares students' answers with correct answers stored in a database and issues an error message if there is no identity with one of them. The set of correct answers is in a database (fully analysed). The students' answers are also fully analysed. Assuming that (a)-(f) are correct answers ('someone learns a foreign language) and (g) the answer by the student:

(a) Man lernt eine fremde Sprache.
(b)Jemand lernt eine fremde Sprache.
(c) Eine fremde Sprache wird erlernt.
(d) Eine fremde Sprache wird gelernt.
(e) Man lernt eine Fremdsprache.
(f) Jemand lernt eine Fremdsprache
(g) Jemand lernt einer fremden Sprache.

The answer (g) is most similar to (b). Similarity is determined by a successive comparison of (linguistic) information. The first information that is compared is word strings. If the words are the same (in the same order) there is identity. If there is no identity with one of the solutions the normalised forms of the words (lus) are compared. (g) and (b) have the same lexical units: Jemand, lernen, ein, fremd, Sprache. So, (g) is most similar to (b). The comparison can take place revealing a difference in case for the nominal elements 'ein', 'fremd'

and 'Sprache'. One could ask whether the kind of checking could not be done by a tool like Hot Potatoes (HP). HP allows for storing the set of correct answers. If the student's answer is identical with one of the stored ones, it is said to be correct. If there is a mistake HP simply says for the first letter that disagrees in the student's answer that from there the answer is wrong and asks for a guess of the next letter. So, if the student gives (a) as an answer, (a) is compared with (b)

(a) *Jemand lernt einer fremden Sprache*
(b) *Jemand lernt eine fremde Sprache*.

the first letter that does not agree between (a) and (b) is the 'r' of 'einer'. HP says: 'r' is an incorrect letter'. This is, of course, not very helpful. 'K-DIFF' represents a much more intelligent solution.

Linguistic richness

'Linguistic richness' is measured by assigning numeric indicators to a text type. The point is to determine factors such as lexical density or grammar complexity. A prerequisite is that there should be a text tagged according to a rich tagset (the more information available, the more linguistic levels can be assessed). Once this information is there searching for indicators that provide relevant evidence can start.

Statistical methods

A statistical analysis can determine parameters like frequency of syntactic category, semantic classes, of category sequences, of collocations, discourse markers, cue phrases, speech act markers etc.. Another way to use statistics is to check texts on a global level and try to detect whether the author applies a nominal style rather than a verbal style. Use of subjunctive may indicate that there is deviation from an 'ideal' text model. Such 'ideal' text models are the prerequisite for having such results.

Information extraction technologies

Another technology to be used in ALLES is based on information extraction. Extraction of information means here finding those terms that optimally characterise the text from a semantic point of view. The processing includes a linguistic analysis (for example the. detection of compound nouns, noun phrases) and complex statistical procedures. The technology is used in automatic indexation and document management to classify documents. One way of using this technology is to check texts for global content properties, such as whether the text is about the topic it is supposed to be by comparing the result of indexation with a predefined (handmade) set of descriptors. If recall and precision are appropriate the student's production can be considered appropriate. The checking excludes the possibility that a learner creates a text well formed from a syntactic point of view, but being about a topic that is not required.

The Combination of Techniques

The different techniques can be combined. The task in ALLES may be to write an e-mail for registering for some training course. Reasons must be given for choosing the specific course. A sequence of evaluation would foresee that first, the correctness of spelling and grammar is checked. The errors must be corrected. Then, a semantic analysis will be applied. Speech act analysis is made then and the errors concerning speech acts are detected and corrected. Information extraction can find most important concepts. If they comply with the concepts to be expected for the task the answer can be rated correct.

The use of language technology for checking learners' input seems to indicate sequential processing starting from the lowest level (spell checking) moving up to the higher levels. This is due to the fact that for deeper level analyses the lower level errors have to be corrected. If an input is to be checked according to specific speech act correctness, this may include semantic analysis and checking of formal properties of the speech act. If there were a number of spelling errors then these words are not available for any of the higher level analyses as long as they are not corrected. The same is true for syntax errors and semantic errors. If the text is full of syntax errors this has an impact on the detection of speech act indicators. As far as the evaluation of 'linguistic richness' is concerned by definition it can be checked only on the basis of correct input.

Other relevant topics

There are a number of topics relevant for 2nd language learning that have not been addressed here. There is e.g. user-orientedness and feedback (self learning systems with 'automatic control' bear a number of questions according to how the interaction with the student is to be designed), testing methodology, software architecture, the issue of e-learning standards and also the use of ASR. ASR is not used in this project as one would expect to check phonetic capabilities of students, but simply to make transcriptions from oral productions by students that are then processed by the NLP tools. Problems resulting from this are either resolved by limiting the possible input dramatically or by specific measures such as having supervised training of the speech tool or even mixing the student's speech with that of a native speaker.

Though these problems have been ignored in this paper, they are not ignored in the project itself. The focus of this paper was simply meant to be on the use of NLP tools in 2nd language learning.

References

Carl, Michael, Antje Schmidt-Wigger & Munpyo Hong (1997). KURD - A Formalism for Shallow Post Morphological Processing. In : Proc. of NLPRS'97.

Hirst, G. J. and Budanitski, A. 2001. 'Correcting Real-Word Spelling Errors by Restoring Lexical Cohesion'. In ACL, Vol. 1, No. 1.

Heringer, H.J. (2000). Fehlerlexikon. Aus Fehlern lernen: Beispiele und Diagnosen. Berlin, Cornelson Verlag.

Karlsson, F. *et al.* 1995. *Constraint Grammar: A Language-Independent Formalism for Parsing Unrestricted Text*. Mouton de Gruyter: Berlin/NY.

Kukich, K. 1992. Techniques for automatically correcting words in text, in *ACM Computing Surveys*, 24: 377-439

St-Onge, D. 1995. *Detecting and correcting malapropisms with lexical chains*, Master's thesis, Department of Computer Science, University of Toronto.