ALLES: Controlled language tools' and 'information extraction tools' for CALL 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 !, and 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 applications. It describes how an autonomous, long-distance 2nd language learning system for advanced learners can be created. The architecture of the system consists of a web-based multimodal user interface, skill-specific learning tools (reading, listening, speaking and writing tool), and a set of NLP-based evaluation tools. All modules are integrated in a flexible software architecture ensuring a user-friendly environment based on advanced concepts in language didactics. The thrust of the project is to show the potential of NLP for automatic evaluation of students' productions.

1. Introduction

This paper will show how mature language technology can be adapted to deliver intelligent CALL. The work introduced here has mainly been done in a project called ALLES.1

ALLES is based on an elaborate didactic model. The architecture consists of three major modules a multimodal user interface, a set of skill-specific learning tools, and a set of NLP-based evaluation tools integrated in a flexible and scalable software architecture allowing for the use of multimedia technologies and providing a user-friendly environment.

NLP tools are used to evaluate a) the **accuracy** and b) the **richness** of learners' linguistic output in a self-learning environment. They comprise:

Morphosyntactic taggers supposed to deliver an unambiguous feature bundle for each linguistic item and to detect errors on word level. The feature bundles are input for **specific control modules** that do further checking on the basis of the **KURD framework [1]** which is a formalism specifically tailored to do error checking. KURD allows for actions to be performed on feature bundles given certain conditions. One of these actions is issuing an error message that is linked to a didactics-driven interpretation of the message. This kind of checking represents a specific error-detection strategy applied in controlled language checking. The tool has been tested extensively in other applications.

Speech recognition tools, used only as a means to transcribe spoken input. The output will be handled by the evaluation tools for written text.

Statistical tools, used for measuring the 'richness' of texts. Texts will be analysed with shallow parsers. A set of lexical, grammatical and (numeric) discourse indicators will be extracted. These indicators will be contrasted with threshold indicators extracted from corpora written by native speakers.

Information extraction tools otherwise used for automatic indexation extract terms from a text that represent its content. This set of descriptors can automatically compared with a handmade set to get insights about the contents of texts.

2. Assessment of linguistic correctness

Linguistic checking means correction of orthography, grammatical structure, adequate semantic use of words, and discourse structure. However, at the current stage of NLP, only spell-checking and, partially, grammar checking have been tackled to a sufficient extent [4]. Automatic evaluation of semantics and discourse seems to depend on a full modelling of natural language understanding inclusive of pragmatics, world knowledge and inferential capabilities. This, of course, is not available for unrestricted text so far. ALLES explores how NL checking could be used beyond spell and grammar checking based on pattern matching.

Thus, an important distinction is global versus exercise-specific checking: Global checking is 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.

2.1 Global checking

Global checking is available for spell checking and the major part of grammar checking.

2.1.1 Spell checking:

The most common and reasonably achieved task is spell checking. Usually the procedure consists in detecting all words that are not present in the lexicon, and providing

1 ALLES (Advanced Long-distance Language Education System) is being funded under the European IST programme (IST-2001-34246). Its main objective is to develop an autonomous, long-distance second-language learning system for advanced learners for four languages (English, Spanish, German, Catalan) in the domain of business and economy.

a list of candidates for correction by means of applying a minimum 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 system2. 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 vs. hyphenation) are a nightmare to formalise).

2.1.2 Grammar checking:

ALLES has a rule-based approach to grammar checking. Recent statistical approaches were either domain dependent or not very successful [2], [5].

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.

The approach taken for ALLES is to directly search for specific grammatical errors and not rely on an a parsing that is based on a concept of well formedness. How this spells out in detail will be described in the next section.

2.2 Restricted NL-checking

As modelling of full language understanding is not available,. the checking of semantic and / or pragmatic well-formedness seems problematic. There are two alternatives: 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 very small text corpus. ALLES confines this sort of NL checking to specific exercises which provides the required constraints and thus concentrates on the first alternative. In addition, ALLES uses simple pattern matching techniques, no deep analysis.

2.2.1 Semantic checking

For speaking a foreign language the appropriate choice of words in a given context is important and creates specific problems for foreign language learners. Automatic checking of this kind of errors would be a huge progress for CALL. Heringer [3] gives an extensive account of this sort of errors (mainly interferences) made by learners of German.

E.g. the two German verbs ‘annehmen’ (accept, adopt), and ‘adoptieren’ (adopt) are very difficult to separate above all for speakers of Germen with an English of French background. The first reading denotes the more general concept of ‘accept’ (in this sense one can ‘annehmen’ (accept) a present, anything that is offered to you), but also a very special concept of adopt in the sense of ‘adopt a decision / a bill / a law’. The second verb is very special, namely used only in the sense of ‘to adopt a child’. For learners of German the use of the two verbs is a real problem. You can frequently hear ‘adoptieren’ instead of ‘annehmen’ from English (or French) native speakers:

*(a) *Der Mann adoptierte den katholischen Glauben.*

(The man adopted the Catholic religion)

(b) Der Mann nahm den katholischen Glauben an.
(The man adopted the Catholic religion)

*(c) *Das Parlament adoptierte den Beschluss.*

(The Parliament adopted the decision.)

(a) is not well formed in German neither is (c). These interferences, though, are quite common. An automatic checking could look if there is a human direct object in a German sentence. Only then it is guaranteed that ‘adoptieren’ is correctly used. All other cases can be rejected as not well formed.

There are other such cases. The three German verbs ‚gehören’, ‚gehören zu’, ‚angehören’ are all referring to a different shade of meaning of ‘belong to’:

(a) Das Haus gehörte dem Mann.
(The house belonged to the man.)
(The house is owned by the man.)
(b) Der Mann gehörte der Universität an.
(The man belonged to the university.)
(The man was a member of the University.)
(c) Der Mann gehörte zu der Universität.
(The man belonged to the university.)
(The man was a member of the University.)
*(d) *Das Haus gehörte dem Mann an.*
(The house was a member of the man.).

Leaving aside the figurative reading of (b) (saying that the man is ‘eaten up’ by his work at the university) we have the following facts: ‘gehören’ with dative denotes ownership, ‘gehören zu’ and ‘angehören’ denote membership.

Valency, though, not globally available can be provided for a limited vocabulary. Our morphological analysis provides lexical semantic information. ‘Haus’ is a ‘location/agent’, ‘Mann’ is an agent, ‘Universität’ is agent / building. (d), thus, can definitely be excluded. A checking would take into account that ‘angehören’ needs a kind of ‘collective object’ (agent / location) as an indirect object.

The general strategy for handling this example is to use lexical semantic information and pattern matching techniques.

2.2.2 Speech acts

A second area that will tried to be modelled with pattern matching techniques is the formal correctness of speech acts.

The task in an exercise may be to reject a request:

Could you please give me your car?

[2] Notice that many proper nouns, abbreviations, etc. are absent in most lexicons.

Among others there are the following correct answers:

No, I won't.

No, I cannot do that.

No, I can't.

No, I will not give you my car.

Of course, not.

Pattern matching based on syntactic information is possible. It will 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.

2.3 Tools

In this section we will present the formal tool that does the pattern matching job mentioned in the previous sections.

The most important feature of the strategy for error checking applied in ALLES is not to start from the concept 'well formed sentence' and work on from that, but to detect the type of error directly. This is to encode the expected errors and try to find it. The strategy has been tested in other contexts, especially for controlled language checking.

This means on the other hand that not ANY error will be detected and commented with a sensible error message, but only those that can be expected to occur for a specific audience.

There are two tools that do the (matching) job. One is called KURD [1], the other one K-DIFF. The first one does pattern matching on the basis of linguistic representations, 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 the 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 an article: feminine, singular, (dative or genitive), masculine, singular, nominative or plural, genitive. 'der' is also a relative pronoun, again with several readings.

KURD allows to define patterns to be mapped on such representations. If the mapping is successful, modifications of the features can be done. The formalism has elements of unification frameworks, but is much more powerful. (KURD: kill, unify, replace, delete). A KURD rule consists of a description and an action part. The description consists of conditions that must match, successive feature structures representing (sequences of) words. They are marked for modification in the action part. A rule fails if the set of conditions does not match. It is executed if all conditions apply.

An example that contains already an error treatment will explain this basic mechanism:

A frequent error for learners of German is the placement of commas in front of a coordinator (which is quite usual, e.g. in English, but not in German). The following rule detects this kind of error:

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

There is a condition part that says, if there is a comma (1st condition) followed by an 'und' (and) or an 'oder' (or) (2nd condition) and anything (this is the lexical item that follows the coordinator) that is 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 in this case saying 'Wrong placement of comma'.

The second tool that is used for automatic error detection is K-DIFF. It compares input with correct answers stored in a database and issues an error message if there is no identity with one of the items. The set of correct answers (linguistically analysed) is stored in a database. The students' answers are also analysed. (a)-(f) are correct answers ('someone learns a foreign language) and (g) may be 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) determined by a similarity algorithm. Similarity is determined by a successive comparison of (linguistic) information. First, word strings are compared. If the words are the same and occur in the same order then this is identity. If there is no identity, the normalised forms of the words (lexical units) are compared. (g) and (b) have the same lus: Jemand, lernen, ein, fremd, Sprache. So, (g) is most similar to (b). The comparison reveals a difference in case for the nominal elements 'ein', 'fremd', 'Sprache'. The K-DIFF approach goes far beyond simple checking of identity as provided by a tool like Hot Potatoes (HP). If the student's answer is identical with one of the stored ones in HP, it says 'correct'. If there is a mistake HP simply says for the first letter that disagrees in the student's answer that from this letter the answer is wrong. This is not at all intelligent.

3. Linguistic richness

The determination of 'linguistic richness' identifies factors such as lexical density or grammar complexity. A prerequisite again is a text tagged according to a rich tagset. Once the information is available searching for indicators that provide relevant evidence can be done.

3.1 Statistics

A statistical analysis can determine parameters like frequency of syntactic category, frequency of semantic classes, specific category sequences, collocations, discourse markers such as cue phrases, speech act markers and others. Statistical results also can check texts according to stylistic properties, e.g. according to nominal or verbal style. Many of these parameters may indicate that there is a deviation from an 'ideal' text model. Such 'ideal' text models on the other hand are a prerequisite for determining such results by comparison.

3.2 Information extraction tools

Another sort of tools that are used in ALLES are information extraction tools. Extraction of information here means 'indexation' which is the extraction of terms from a text that characterise it semantically. The processing consists of a flat linguistic analysis, (especially, detection of compound nouns and noun phrases) and some complex weighting that delivers the wanted terms, called descriptors. This technology is used for document classification. For ALLES it is used to detect global properties of texts, e.g. whether the text is about the topic it is supposed to be by comparing the result of the indexation with a predefined (handmade) set of descriptors. Success is measured in terms of recall and precision. If both are appropriate the student's production can be considered correct. The checking excludes that a learner creates a syntactically well formed text being about a topic that is not required.

The different techniques must be combined: If a task for a student is to write an e-mail for registering for a training course, and if reasons must be given for choosing this course, a sequence of evaluations my happen: First, the correctness of spelling and grammar is checked, that then the errors must be corrected. Then, some semantic checking may be done and speech act analysis. Errors also are then corrected. Information extraction can find most important concepts. If they comply with the concepts to be expected for the task the answer can be considered correct.

The use of language technology for checking learners' input seems to require sequential processing from the lowest level (spell checking) to the higher levels. This is due to the fact that for deeper level analyses the lower level errors have to be corrected. So, if an input is to be checked according to speech act correctness, this may include semantic analysis and checking of formal properties of the speech act. If there were spelling errors then these words would not be available for any of the higher level analyses. 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. 'Linguistic richness' by definition can be checked only on the basis of correct input.

4. Other Issues

A number of other issues are relevant for (computer assisted) 2nd language learning that have not been addressed in this paper, among them there are user-orientedness / feedback (interaction with the student), testing methodology, software architecture, the issue of e-learning standards and also the use of ASR. ASR is not used in this project to check phonetic capabilities of students, but simply to make transcriptions from oral productions which are then processed by the NLP tools. Supervised training for the speech tool or mixing the student's speech with that of a native speaker will avoid problems in this area. Though these problems have been ignored in this paper, they are not ignored in the project itself. The focus of this paper was meant to be on the use of NLP tools in 2nd language learning.

References

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

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

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

[4] Kukich, K. (1992). Techniques for automatically correcting words in text. ACM Computing Surveys, 24: 377-439

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