

A Web-Based System for Automatic Language Skill Assessment:

EVALING

Cédric Fairon

Université Paris 7, LADL (France)

fairon@ladl.jussieu.fr

Cédric Fairon
Université Paris 7
Laboratoire d'automatique documentaire et linguistique (LADL)
2, Place Jussieu (CASE 7031)
75251 Paris CEDEX 05
France

A Web-Based System for Automatic Language Skill Assessment: EVALING

Introduction

EVALING is a Leonardo da Vinci project funded by the European Union. It involves four European laboratories¹. The aim of the project is to build an automatic system to evaluate language skills in the mother tongue. Each partner works on his own language and builds specific tests (at the present time, three languages are at work: French, Italian and German. In this paper we will present French.

The first step consists, for each language, in determining the fields to tested and the types of exercises which can be computerized to carry out this task. The task differs from one language to another. For example, spelling exercises are not very interesting for German, since people make few mistakes, but they take on a full meaning in French. For French, we decided focus on syntax, lexicon, spelling and reading comprehension. Hence, we oriented the reflection on the tests that could be automatized and on those that could not. At this point, automatization in the Eevaling sytem bears on three phases of the evaluation process:

- dynamic setting up of tests (exercises are stored in tables of an exercise database). We assume that if a person has to take the test more than once, this person should not get twice the same set of questions,
- automatic grading of tests (including storage of marks in a client database),
- semi-automatic filling of exercise databases (with the assistance of linguistic tools).

First, we discuss some technical aspects: Eevaling is a Web-based program interacting with large exercise databases and a client database. We will explain how this "item Banking"² system has been implemented on a Web Server as an ISAPI (Internet Server Application Programming for Microsoft Information Server³) and how exercise databases were built. This last point is a key issue, because the aim is not so much to gather a fixed amount of exercises, but rather to be able to renew them easily. We designed a set of linguistic tools to satisfy this demand. The set of tools is based on the software INTEX, developed at the LADL by Max Silberztein⁴. Of course, it is not always possible to automatize the creation of exercises. In certain cases, the work will have to be done manually.

Second, we will present the 'Administrator side'. We call 'administrator' the person or team who needs to evaluate a large group (students, employees, applicants, etc.). An interface enables the administrator to define the parameters of the test (length, level, etc.) and to perform some statistical analysis on the client database.

¹ Association pour le traitement automatique des langues (ASSTRIL) for French, Consorzio Lexicon Ricerche from University of Salerno, for Italian, Pädagogische Hochschule Karlsruhe and Universität München for German.

² For references about "item banking", see Brown (1997), p.43.

³ For further informations about ISAPI programming advantages and disadvantages, see Chapman (1997).

⁴ INTEX is a parsing system based on wide coverage dictionaries and local grammars (represented by Finite State Automata) which applies to large corpora, that is 100Mo. Cf. Silberztein (1993).

Finally, we will deal with the 'client side'. The client registers himself and has then access to the test session through a Web-Client software. All test forms are in HTML form.

1. The Developer side

1.1 The Main Program and the Databases

From a technical point of view, Evaling uses a Multithreaded Automation Object. This technology allows multiple threads of execution (Apartment-model threading). Thus, the Evaling kernel is an 'Active X DLL'. In the DLL, all exercises are designed on a single module which makes it easier to develop and manage the whole system. In fact, in order to add or remove exercises, we just have to add or remove modules and to change a global variable in the main module which indicates to the system the number of exercises and their order in the session. The system is developed to run with Microsoft Information Server.

Sets of exercises passed down to a client are dynamically composed by the system at the time of the user's request. All exercises are stored in tables of the Exercise database. For each questionnaire, the system extracts a set of exercises from a table. Each record in the table corresponds to one sentence, or one text of exercise. Therefore, several records are retrieved by the system to constitute the questionnaire that will be sent to the client. It is imperative that sets of exercises have the same level of difficulty from one client to another. To signal this stability, a field in each record indicates the level of difficulty. When the Evaling system composes a questionnaire, records are chosen in such a way that the sum of the difficulty marks is always the same (this sum is given by the administrator who can choose the global level of difficulty). Grading occurs at the development time. A script, which uses linguistic tools (INTEX programs, local grammars, dictionaries and linguistic descriptions) evaluates the sentences. For French, the difficulty of a sentence depends on:

- length in words,
- presence/absence of negation, conjunction, relative pronoun,
- lexical difficulty: to measure this parameters, we use an electronic dictionary: the DELAF dictionary for French that contains more than 900.000 inflected forms. It is divided into three levels of difficulty: common and easy words constitute the first level, technical and unusual words the third level and in between, the second level is an intermediate part with words whose understanding is hazy.

The dynamical design of a questionnaire requires large databases, with a large number of sentences for each exercise. It is costly to produce all of them manually. To avoid such a difficulty, we designed a set of search tools that apply to large corpora and retrieve sentences or short texts that match a given linguistic pattern. Search tools are generally Finite State Transducers using morpho-syntactical information provided by wide coverage electronic dictionaries. To this end, we call for the framework of LADL tools and data⁵.

Corrections of the user's answers are provided by JavaScript programs that use data of hidden fields in the HTML form or by more complex programs running on the server when necessary. When we make use of JavaScript programs to correct a form, we never display answers clearly in the JavaScript source

(functions that take ASCII values or binary values evaluate the validity of answers). Answers are therefore not readable in the HTML source code. This practice is a second level of protection, because the whole test session occurs in a Browser Window that does not contain the menu bar that allows the view source action.

1.2 Sample of exercise developpement

We present a simple case of exercise building. The purpose of this exercise is to test the ability to choose the right form of the French word “ tout ” (‘all’) that can be noun, pronoun or adverb and that can be spelled “ tout ”, “ tous ”, “ toute ” and “ toutes ”.

1.2.1 Retrieving sentences with a graph

We design a graph and apply it to a corpus to retrieve sentences that match the sequences described by the graph. Then the graph is stored in the “ tool folder ”. It will be reused later, when we will need to refresh the Exercises database.

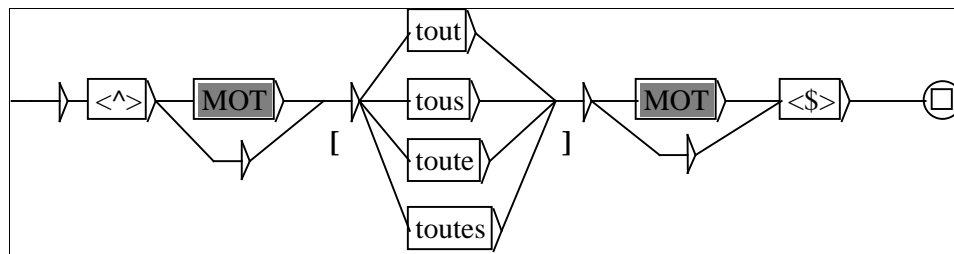


Fig. 1: Sample of locating graph

The graph is read from left to right (from the Initial node to the terminal one). If a sequence of words in the text is matched by one of the paths of the graph, the sequence is saved. An output is associated to this graph (represented below two boxes: “[” and “]”). We apply the graph in merge mode to insert this output in the text. Inserted signs are helpful to import sentences in the database.

Our corpus is preprocessed by INTEX. Since preprocessing segments the text into sentences, it is possible to refer to the “ beginning of a sentence ” (<^> in our graph) or to the “ end of a sentence ” (<\$> in our graph). The box containing “ MOT ” is a link to another graph that represents a string of words (max. 10 words) and optional delimiters (apostrophe, comma, hyphen). This string is optional, that is why there is an alternative path under the box “ MOT ”.

The following sentences are retrieved from a novel of Agatha Christie:

Son air très anglais avait **[tout]** pour séduire quelqu'un qui, comme moi, n'avait pas revu sa patrie depuis trois ans.
 Je voudrais que, de retour chez vous, vous observiez le monde nouveau de l'après-guerre et que vous décidiez, en **[toute]** liberté et indépendance, ce que vous en attendez.
 J'ai fait **[tout]** mon possible pour ne pas vous dire que je vous aime...
 "Et ils vécutent **[tous]** ensemble dans une petite maison biscornue.

⁵ For a description of dictionaries format and samples of application, see Courtois (1990).

of difficulty to compute. Since all marks are stored in a database, it is possible to perform some statistical analysis. Through the Administrator Interface, one can handle the usual numerical data: who got highest/lowest marks, where are the weaknesses/strengths of the group, etc. Statistical information is then displayed graphically and in tables. Example of application: In October 99, at the beginning of the academic year, EVALING will be used to help a language teacher with information about the specificity of this group of students. A statistical analysis of the marks obtained by about two hundred students is already a rich information basis. The administrator can detect common weaknesses and adapt his course to these difficulties.

3. The User side

The test session occurs in a Web Browser Window (a simple window, without Toolbar and Menubar). First, the user has to register. After submitting his registration form, he receives a User Key that allows him to enter the test session. This user key may be reused later if the client has to pass the test for a second time (registration occurs only one time). Once he has registered, the user clicks the button "Beginning" and receives from the server a first set of exercises (e.g. exercises on noun phrases). He answers by filling the HTML form (pop-up menu, text area, radio-button and checkboxes). When this is done, he asks for the next exercise by clicking on a button. The correction is done by the JavaScript (sometimes the correction occurs on the server) and a mark is sent to the system and stored in the client-database. Then the system returns the next exercise (e.g. exercises on past participles) to the client. At the present time, the test session is not "adaptive"⁶: all clients answer the same questions (layouts used to compose sets of questions are the same for all users). At the end of the session, the client receives a report, in case the Administrator has configured the system to do so. The report mentions only the marks registered by each exercise, without any other form of comment. Marks are also represented graphically by a Java Applet.

We were careful to avoid interferences between language skills and computer skills. Computerization should not interfere with language skill assessment. In the EVALING system, it is a prerequisite that a user non familiar with computer has to carry off easily and successfully the computer handling necessary to take the test. Recent scientific studies showed that there is little or no evidence of adverse effects on the computer-based tests due to lack of prior computer experience⁷. To ensure that, we decided to rule out technical manipulations like "drag and drop" and to make every effort to build a simple and intuitive interface. A tutorial is available from the start page on. Before getting into the test, the user can discover the different types of questionnaires and try each element of the answering system (in fact, any item in HTML Forms: pop-up menu, text field, button, radio button, etc). During the test a help button always displayed at the same location provides a contextual information by means of a simple click.

⁶ Principles of Computer-Adaptive language Tests (CAT) are clearly described in Tung (1986).

⁷ Taylor (1998) for the TOEFLE study and Russell (1997) for a comparison on multiple-choice tests.

Conclusion

The originality of our work lies in the use of powerful linguistic tools that can be adapted to a large variety of situations and that allow easy, fast and cheaper renewal of the stock of exercises (with the aim of changing levels of difficulty or testing skills in technical languages).

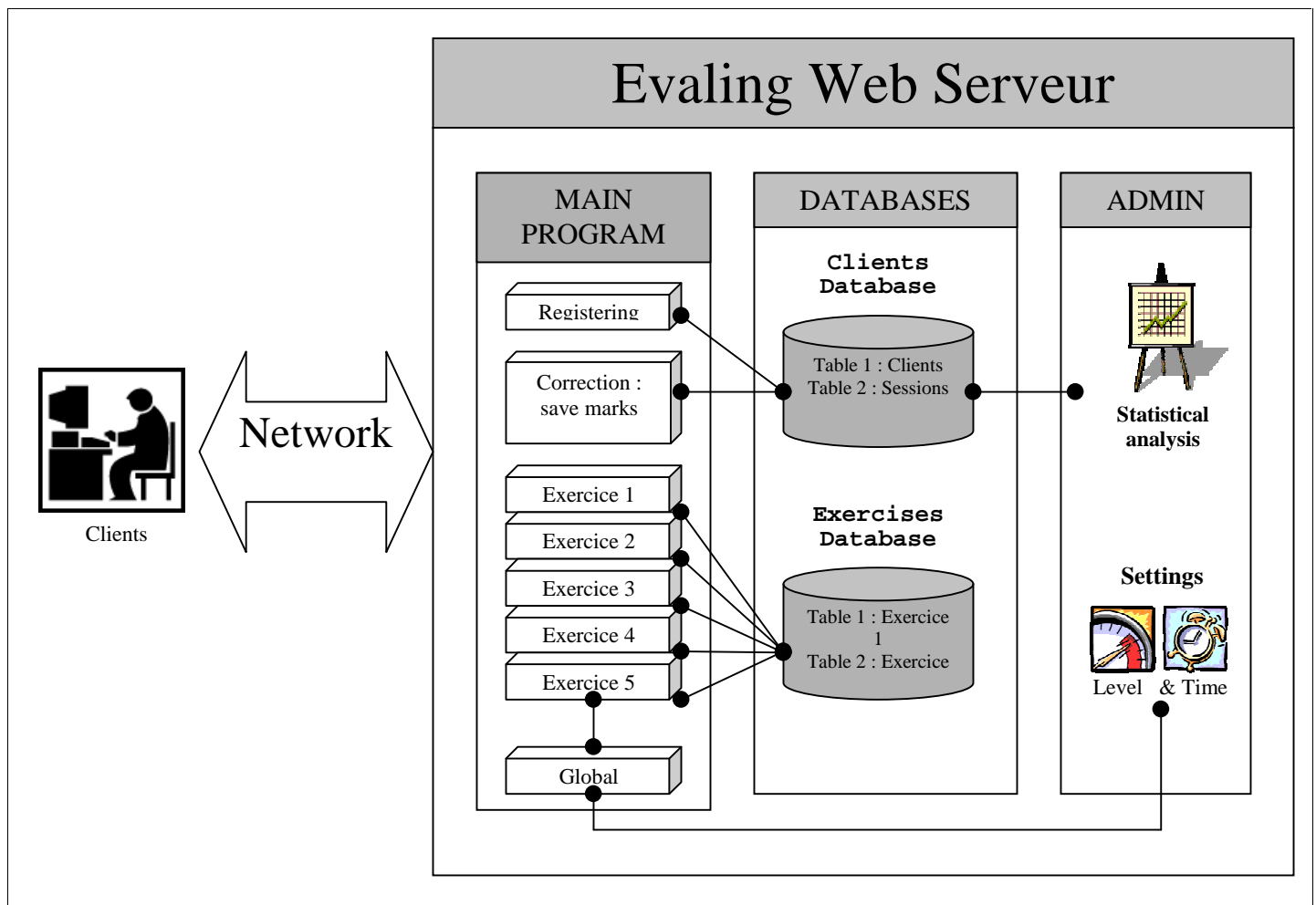


Fig. 4: The Evaling system

References

- SILBERZTEIN M. (1993). *Dictionnaires électroniques et analyse automatique de textes. Le système INTEX*. Paris: Masson.
- COURTOIS Bl., SILBERZTEIN M.(ed.) (1990). *Dictionnaires électroniques du français. Langue française. N°97*. Paris: Larousse.
- BROWN, J.D. (1997). *Computers in language testing: present research and some and some future directions*. In *Language learning & technology*. Vol.1, N°1. (on-line access: <http://polyglot.cal.msu.edu/llt/>).
- TAYLOR C., JAMIESON J., EIGNOR D., KIRSCH I. (1998). *The Relationship Between Computer Familiarity and Performance on Computer-based TOEFL Test Tasks*. Princeton: Education Testing Service.
- RUSSELL M., HANEY W. (1997). *Testing Writing on Computers: An Experiment Comparing Student Performance on Tests Conducted via Computer and via Paper-and-Pencil*. In *Education Policy Analysis Archives*. Vol. 5. N°3. (on-line access: <http://olam.ed.asu.edu/epaa/arch.html>).
- CHAPMAN D. (1997). *Wed Development with Visual Basic 5*. Indianapolis: Que Corporation, MacMillan.
- TUNG, P. (1986). *Computerized adaptive testing: Implications for language test developers*. In *Technology and language testing*. Washington, DC: TESOL.