# Italian Arabic linguistic tools

**Eugenio Picchi, Eva Sassolini, Ouafae Nahli,
Sebastiana Cucurullo, M. Isabel Vargas**

Istituto di Linguistica Computazionale
Consiglio Nazionale delle Ricerche
Via Moruzzi 1, Pisa, Italy
picchi@ilc.cnr.it

**Abstract**

This paper concerns our participation in the research project: 'Corpus bilingue Italiano *Bilingual Italian – Arabic corpus)* funded by law 488/92. The purpose of this project is to develop some linguistic tools and resources for bilingual Italian/Arabic corpora; its background and starting point are tools that have already been developed by the Computational Linguistics Institute. As far as IT tools are concerned, the project consists of four basic elements: a) morphological engine for the Arabic language; b) *aligning* system for Italian and Arabic parallel texts; c) automatic *tagging* system for Italian and Arabic texts; d) access tools (and relevant *query systems)* for the texts of the bilingual corpora at each text-processing step.

## Introduction

In the framework of the comprehensive "Linguistica Computazionale: ricerche monolingui e multilingui" (*Computational Linguistics: monolingual and multilingual research*) project funded by law 488/1999, the Istituto di Linguistica Computazionale has taken part in the study and development of tools and resources for the Arabic language, as part of the "Corpus bilingue Italiano – Arabo" (*Italian – Arabic bilingual corpus*) objective. This objective involves the development of a bilingual linguistic work environment, consisting of Italian and Arabic tools and resources, with special attention to the contrastive aspect of it.

Bilingual corpora are innovative researching tools that work by comparing relevant languages and/or cultures, that are essential to develop computer-assisted teaching methods and acquire most of the knowledge on which the development of the most promising multilingual IT applications is based (translating aids, information retrieval, data mining, etc.).

The objective has been developed in co-operation with the Istituto Universitario Orientale of Naples and the "Dipartimento di Scienze Storiche del Mondo Antico" of Pisa University, which have taken care of developing its linguistic aspect, while we developed all its software features.

*Linguistic Tools*

Textual analysis procedures
Morphological engines
Taggers
Aligner

*Linguistic resources*

Monolingual reference corpora
Automatic lexicons
Bilingual aligned corpora
Tagged corpora

As a *background* contribution, the Istitituto di Linguistica Computazionale provided the PiSystem, an integrated linguistic analysis system developed by Eugenio Picchi, which has become the standard for many projects based on the study and analysis of different types of texts, and the basic engine of which is the DBT (Data Base Testuale – *Textual Data Base*) system for the analysis and use of textual resources. The PiSystem features used in the project were its existing Italian modules, such as PiMorfo (Italian morphological engine), PiTagger (automatic Italian morpho-syntactic disambiguator) and Synchro (procedure for the automatic "synchronisation" of parallel texts, already used in Italian-English and Italian-Latin bilingual applications). In addition, such tools have been the basis for the development of matching features in an Italian-Arabic bilingual system.

The project in its entirety involves the development of some linguistic resources:

- generic corpus (8 million words)
- aligned parallel corpus (4 million words)
- tagged corpus (2 million words)
-morphological lexical resources (20,000 entries)

## The Arabic textual analysis system and relevant *"query system"*

The 256-type encoding system provided by ISO 8859-6 (Arabic) charset has been used all through the project, for potential interchange with other partners, acquisition of existing texts and materials, and development of software tools.

The Arabic alphabet is composed of 28 letters, which are differently shaped depending on their position (initial, middle, final or isolated), since these letters have to be linked to each other (except a group of six letters) to make words. Extremely important was the decision to adopt one encoding system as much for the acquisition and entry of linguistic materials as for internal representations and processing. Due to the bilingual nature of the project and with a view to being able to use the materials and tools independent of the availability of native Arabic computers and operating systems, the strategy chosen was to develop a proprietary system for the interaction with Arabic materials, i.e. a system that can be interactively used through the keyboard and that gives a correct representation, event without using a specialised Arabic

computer or operating system (the development environment is *Windows*).

The keys on the keyboard have been made to match the Arabic alphabet, by selecting it in a manner that matched a standard Arabic keyboard (Fig. 1). Each program was provided with a double function: the above-mentioned matching of the keyboard for normal typing, and the development of a virtual keyboard to be worked on with the mouse to compose a text, queries in particular.

The DBT (Data Base Testuale - *Textual Data Base*) system was the basic tool used in the Arabic language project. Such system, however equipped to manage a whole series of non-Latin alphabets, required substantial changes in order to properly work on Arabic texts. It can display all or part of the text, search words, calculate frequencies, define research functions with several words associated in different ways using logic operators, and retrieve all the contexts that fulfil specific search conditions, generate orderly concords, define specific conditions for concord generation, search by regular phrases, etc.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| q = ق | w = و | E = ا<br>E = ء | R= ع<br>r = ر | T = ط<br>t = ت | Y = ى<br>y = ي | U = ؤ<br>u = ئ | I = ئ<br>i = ِ | O= ئ<br>o = ه |
| A = ا<br>a = ئ | S= ث<br>s = س | D = ض<br>D = د | F = ِ<br>f = ف | g = ج | H = ح<br>h = ه | J = ع<br>j = ذ | K = خ<br>k = ك | l = ل |
| Z = ظ<br>z = ز | x = ش | C = ء<br>C = ص | V = أ<br>v = إ | B = ئ<br>b = ب | n = ن | M = ة<br>m = م | | |

Figure 1: data-entry keyboard

The Arabic-alphabet DBT version keeps the characteristics of such language (such as the text displayed from the right- to the left-side), has been instructed through special descriptive tables on how to read the input text encoding: both for a proper display on screen and in print, and for the determination of its proper alphabetic order. These resources have been designed to comply with ISO-8859-6 standard.

## Morphological engine

The morphological engine has been designed to perform a double function: on one side, to generate the inflexion and, from one Arabic entry, automatically generate all its forms (including the their morpho-syntactic classification), while, on the other side, to allow the morphological analysis, that goes back from one form to the entry (or entries)

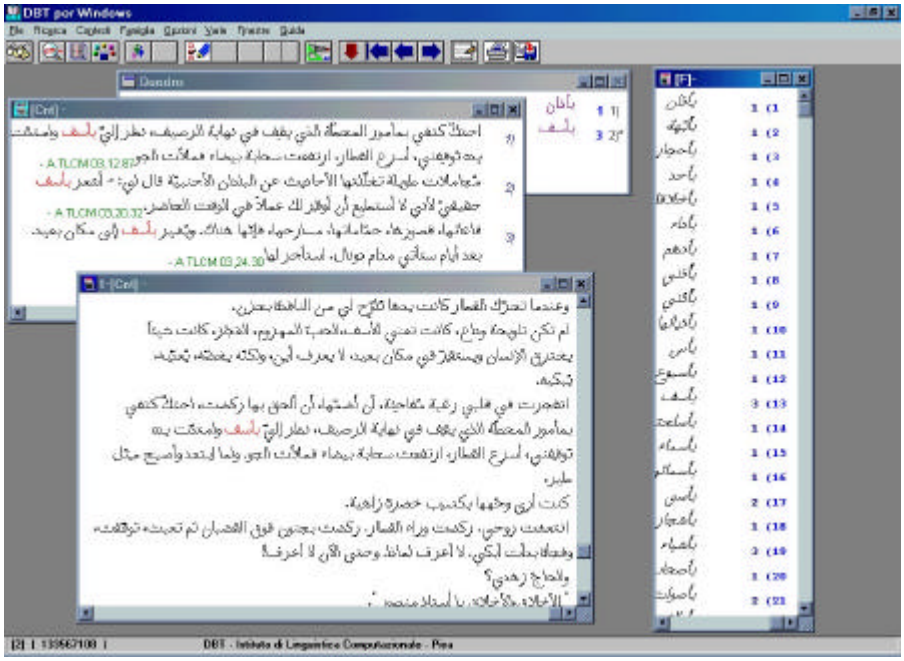


Figure 2: working session using Arabic DBT query system

to which such form belongs, as well as identify its potential, theoretically valid, morpho-syntactic classifications.
To develop such component, we had to:

1. Define the encoding system to be used for a representation of lexical data; definition of the composition, dimension and structure of the Lemmario (entries dictionary); definition of the encoding system, syntax and structure of the "morphological rules" file;
2. Identify groups of entries having the same morphological behaviour and draw up morphological rules based on defined encoding and syntax;
3. Develop a "Lemmario " file and enter suitable inflexion codes in there.
4. Develop software modules for the development and management of supporting files (lemmario and inflexion rules);
5. Develop software modules for generation and automatic analysis;

The grammatical structure is composed of the following:

1. Verbal entries
2. Non-verbal entries:
   - Nouns (that in Arabic include adjectives as well),
   - Relation-words.

**Verbal entries**

Verbal entries are identified by recognising:

- Form active / passive
- Tense completed (or perfect) / uncompleted (or imperfect)
- Mood indicative / energetic I, / energetic II / subjunctive / apocopated / imperative, imperative energetic I, imperative energetic II.
- Gender masculine / feminine / common (masculine/feminine)
- Number singular / dual / plural
- Person first person / second person / third person

Overview of the syntactic structure of verbal entries:

- Triliterals
- The first form
  - Regulars
  - irregulars : geminate verb / verba hèmzata / weak verbs / double irregular
- The derived forms
- Quadrilaterals
- The first form
  - The derived forms

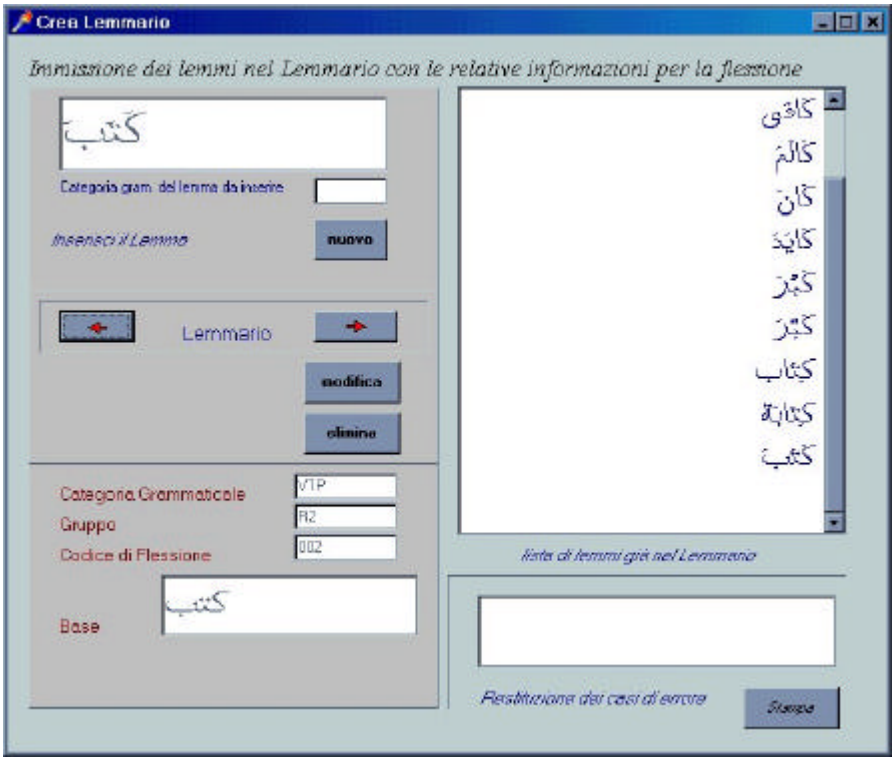


Figure 3: program for the development of the lemmario

**Non-verbal entries**

Noun – substantive / noun – adjective / personal, demonstrative, relative, interrogative pronoun.
Gender: Masculine Nouns / Feminine Nouns / masculine./feminine Nouns / Comparative Adj.
Definition of Nouns : declinable / indeclinable
Declension of Nouns : solar declension / lunar declension
Type of plurals : sanus / fractus / sanus + fractus

Description of main software modules

The functions of the software features will be only briefly described here, since they need to be tested and checked before their final implementation.

The structure and interactions of each component can however be summarised as follows:

Program for the development of the Lemmario. This program manages a mechanism for the listing of an entry word in the "Lemmario" file, inclusive of vowels and all the information required for its processing: grammatical category, inflexion code, etc. The Lemmario tool thus developed will be used by the following software components both during generation and analysis.
Components for the management of the user interface (listing of entry words and retrieval of results), as regards both the generation and analysis portions. The components that manage the mechanisms of the morphological engine proper will be added to such interface.

Generation module

The mechanism used for the entry of types is the same as that used in the software module that develops the Lemmario. The use of the keyboard is the same, and vowels have to be entered for the program to work properly; then, in a later version, the program will also accept entry words entered without vowels.

1. The first step manages the data entry and then checks if the entered word is already contained in the Lemmario, and informs the user thereof;
2. The next step, based on the rule of inflexion associated to the typed entry word, retrieves (if required) prefixes and suffixes from the suitable tables following the steps contained in the rule.
3. As the forms are formed, the bases are created for the inflexion of the different verbal tenses and inflected forms obtained, in case of hamzaed or weak entries.
4. Once the data have been obtained, the program compiles a list of forms that are subsequently processed through special procedures for on-screen display. In addition, the information associated to the entry in the Lemmario is also supplied.

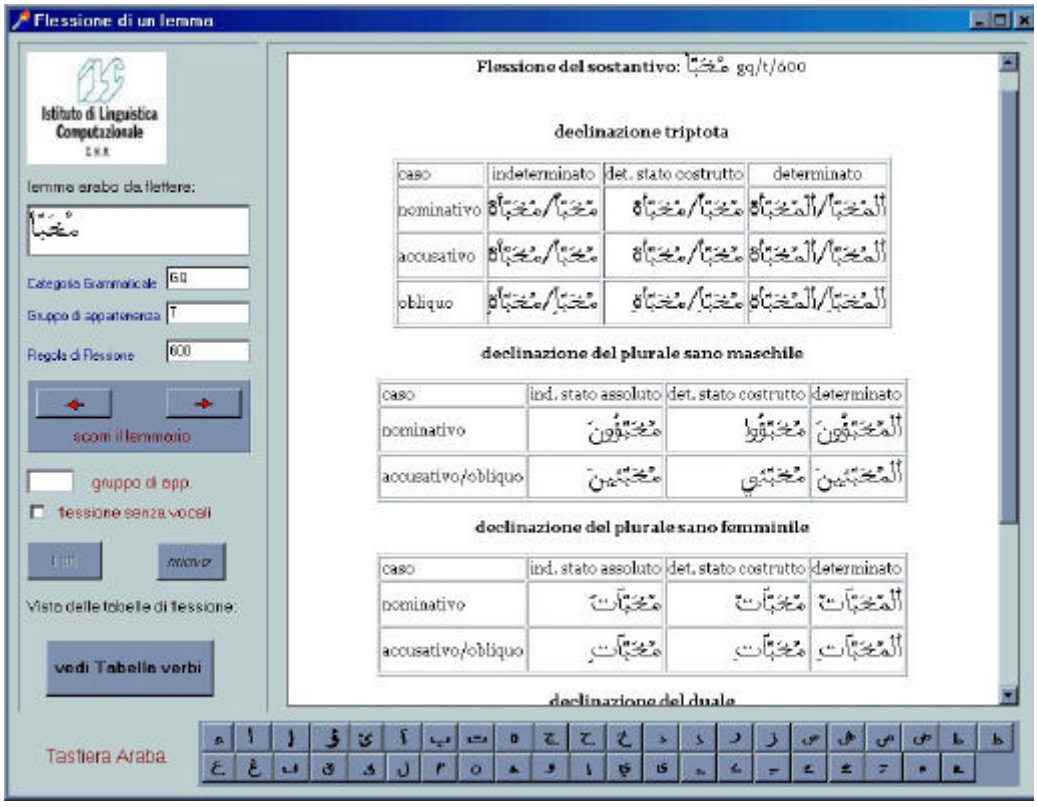


Figure 4: generation program template

Analyser module

1. Any prefix and suffix has to be recognised in order to assume a potential base and search for its recognition within it, through the inflexion of the entry to which the base refers. The mechanism must consider that the form may be lacking in one, two or all the vowels of which it is phonetically composed and still provide for its recognition through recursive search procedures.

2. Search within the Lemmario of the entries associated to the assumed bases.
3. Check, through internal inflexion, that the form belongs to the specific assumption. Development of a list of entries to which the form may belong. The assumptions supplied are more when the key vowels are missing.

Examples of use of the form-generating program from one specific entry (figure 4) and of the text-analysis program (figure 5).

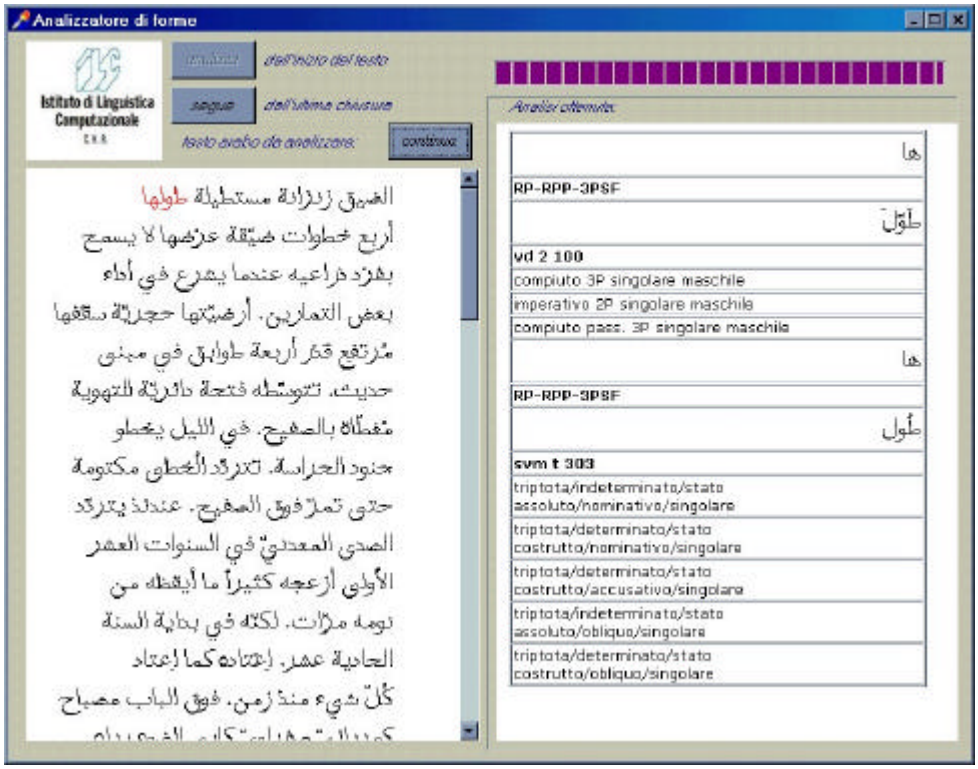

Figure 5: analysing module template

## Arabic text tagger

For the development of the Arabic component, the approach was by matching the *PiTagger* component of the PiSystem system, that in the disambiguation phase uses a statistic approach to select the reference entry and the correct grammatical category of each text word from all those proposed for the morphological component.
A number of integrated components are assembled to build up the entire classing procedure:

- *PiMorfo*: the Arabic morphological engine that, making use of its Arabic lexical system, analyses each text word, relating it to all its potential entries and supplying the relevant grammatical classifications for both the entry and its form.
- *TaggTree*, which is used to process the texts of the *Training Corpus*, to statistically summarise its linguistic behaviour and store the analytical data obtained in the reference database.
- *PiTagger*, a module in charge of processing the text, already morphologically analysed by the analyser, and of automatically disambiguating all those cases in which several alternate solutions have been proposed; such module works on the reference databank.
- *TaggHand*, a module that interactively checks the results of the automatic PiTagger operation and corrects its errors, if any.

The procedure flowchart consists therefore of the following steps:

1. Drawing up of a reference database from the available *Training Corpus*.
2. Morphological classification of each new text to be analysed, using the *PiMorfo* module to associate each word to all its potential lexical and grammatical alternate options.
3. Application of the *PiTagger* module, that automatically disambiguates any ambiguous cases;

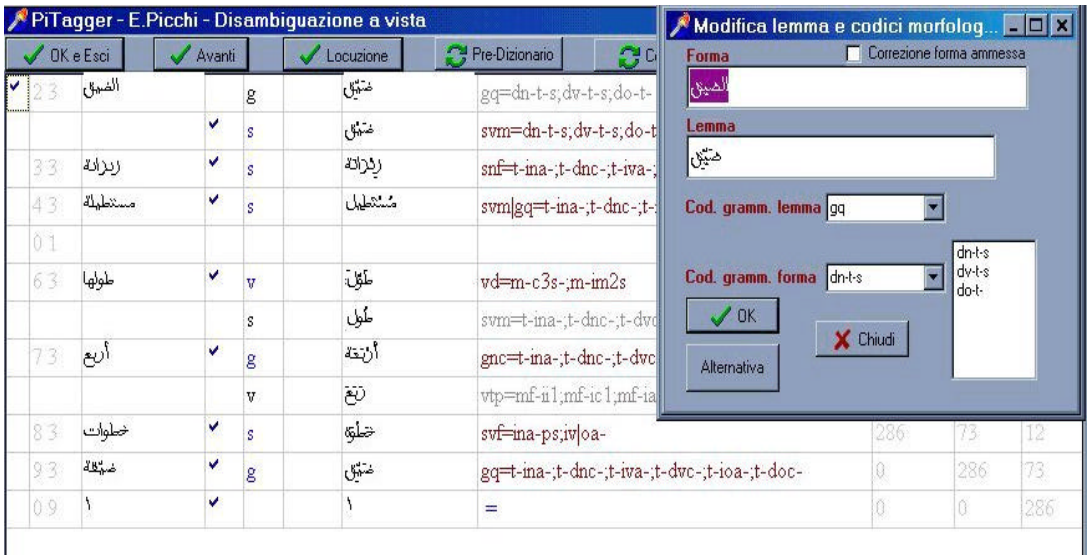


Figure 6: "PiTagger" program template

The procedure flowchart consists therefore of the following steps:

4. Drawing up of a reference database from the available *Training Corpus*.
5. Morphological classification of each new text to be analysed, using the *PiMorfo* module to associate each word to all its potential lexical and grammatical alternate options.
6. Application of the *PiTagger* module, that automatically disambiguates any ambiguous cases;
7. Check of resulting data and correction, if required, through the *TaggHand* procedure, resulting in the generation of the final text, grammatically listed in a dictionary and tagged;

The listed text is available for all the new analysing and querying functions and it can also flow back to the *Training Corpus* to enrich the reference databank and thus make the entire procedure more efficient and productive.

### Aligner – Alignment of parallel texts

The next step involves the automatic alignment of Italian and Arabic parallel texts one being the translation of the other; and enables the system to query the texts in both languages, resulting in the alignment of the two texts.

A procedure has been adopted that implements an aligning algorithm for parallel texts, that, as we mentioned before, make up databanks of parallel texts and represent textual sets composed of texts in some source language L1 and of matching texts translated into a target language L2.

The method used is based on Gale & Church's algorithm implemented by the "Parallel-DBT". This statistic approach is exclusively based on the punctuation and paragraphing used in the texts, regardless of the semantic contents, morphology and syntax of the languages considered. The algorithm does not require, therefore, the use of lexical or morphological aids, dictionaries, grammar rules, inflexion tables, etc.

The purpose of the aligner is to identify matches between sentences in one language and sentences in their translation. The procedure is exclusively based on a statistic model, the main subject of which is the length of the two texts and relevant textual units. The approach proposed by Church e Gale is based on two fundamental principles:

1. Very long sentences in one language tend to be translated into equally long sentences in the other language, and short sentences in the former language tend to be kept short in the latter as well.
2. Some types of alignment are more frequent than others, for instance the occurrence of a 1:1 sentence match recurs a far higher number of times than a 2:2 match or other potential alignment cases.

This is why the algorithm divides up each text into sentences or pericopes (the so-called *soft regions*); then, it comparatively analyses the two texts, working in a sequential order and establishing matches between the *soft regions* of the two texts, using a probabilistic index which is essentially based on the length-based features of the relevant texts.

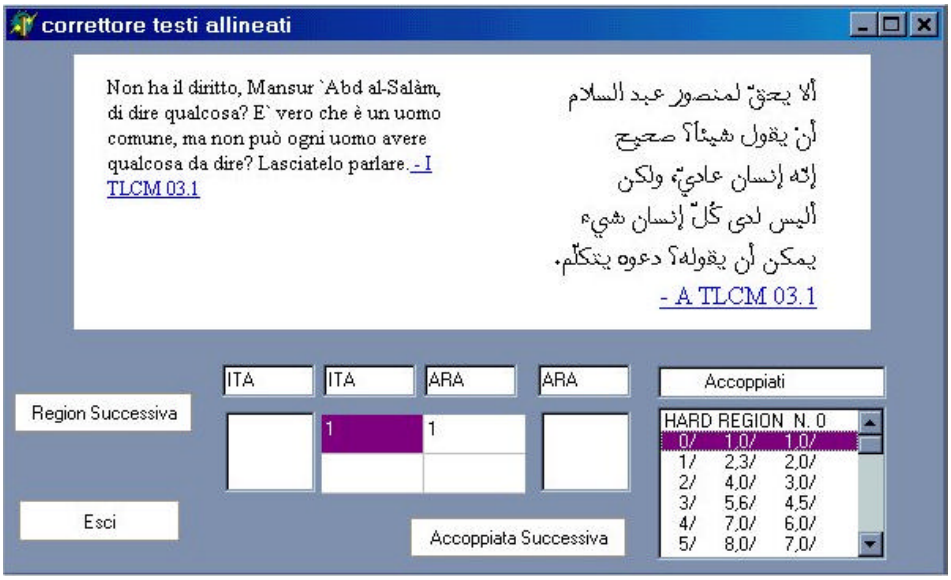


Figure 7: "Parallel-DBT" program template

A *post-editor* (figure 7) can be used to display, assess and change, if required, the results of the alignment obtained through the automatic procedure in order to obtain even better results.

The bilingual search system allows the user to work on each text using the specific context search function of the *DBT query system*. Bilingual files can thus be consulted to search contrastive parallel contexts for equivalent texts in both languages.

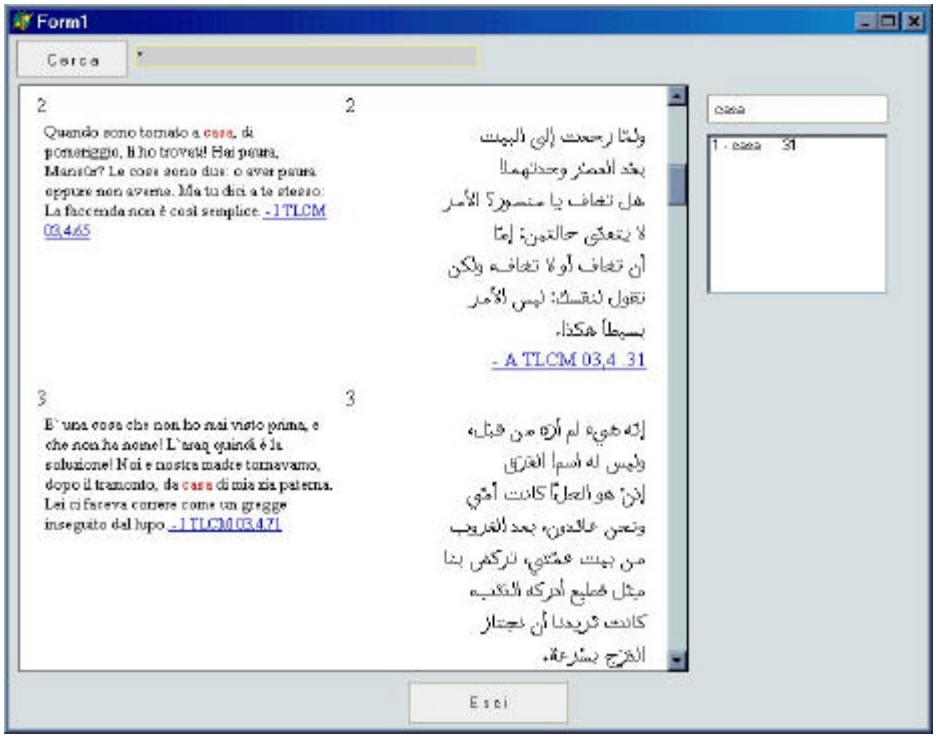


Figure 8: DBT Query System DBT for parallel texts

The results of the alignment operation are filed in the textual *database*, so that this information can reused in the next processing operations, specially for the *query system* that can be used for the contrastive consultation of bilingual corpora.

The searched word does not have an associated *link* of its own that makes immediate reference to the word or matching part of the text, as a translation into the *target* text of the word searched within the *source* text. Using the searching functions provided by the DBT system for the search of words and linguistic elements in general, all the features of the *transfer* mechanism (figure 8) can be observed in the evidence provided by the bilingual corpora. In particular, such tools can be used for bilingual lexical elements, for searching real, proven translations of technical terms and neologisms, for providing more accurate and substantiated information on the behaviour and proper meaning of the rendering from one language into another.

## References


Ballim, A., (1995) - "Deliverable 2.5.2 Aligner v0.2", in Multext Project of March – 1995

Church, K.W., Gale, W. (1991) – "Concordances for Parallel Text" - Using Corpora, Proc. 7th Annual Conference of the UW Centre for the New OED and Text Research - Oxford: OUP, 40-62 – 1991

Church, K.W., Gale, W. (1993) – "A Program for Aligning Sentences in Bilingual Corpora" – Computational Linguistics, 72-102 – 1993

Hartmann, R.R.K. (1994) – "The Use of Parallel Text Corpora in the Generation of Translation Equivalents for Bilingual Lexicography", in Euralex 1994 Proceedings, Amsterdam, 291-297 – 1994

Marinai, E., Peters, C., Picchi, E. (1990) – "The Pisa Multilingual Lexical Data Base System", in Esprit BRA 3030. Twelve Month Deliverable, ILC-ACQ-2-90 – 1990

Marinai, E., Peters, C., Picchi, E. (1991) – "Bilingual Reference Corpora: A System for Parallel Text Retrieval", in Using Corpora, Proc. of 7th Annual Conference of the UW Centre for the New OED and Text Research. Oxford: OUP, 63-70 – 1991

Marinai, E., Peters, C., Picchi, E. (1994) – "A Prototipe System for the semi-automatic sense linking and merging of mono-and bilingual LDBS", in Research in Humanities Computing, ed. by N. Ide and S. Hokey, OUP Oxford – 1994

Peters, C., Picchi, E. (1995) – "Capturing the comparable: a system for querying comparable text corpora", in Computational Linguistics - 1995

Picchi, E. (1991) – "D.B.T. : A Textual Data Base System", In Computational Lexicology and Lexicography, Special issue dedicated to Bernard Quemada, II Ed., Linguistica Computazionale - 1991

Veccia Valieri, L. (2000). Grammatica Teorico-Pratica della lingua Araba. Roma, Istituto per l'Oriente vol Ie vol II, 2000

Veccia Valieri, L. (1992). Complemento della morfologia e sintassi, Roma, Istituto per l'Oriente, 1992

Zerboni, F. (1998 – 2001). www.sit5.com/recensioni/software.

Paragon Software (Smart Handheld Devices Division) (1998 – 2002). www.penreader.com