

Multilinguization and Personalization of NL-based Systems

Najeh Hajlaoui

GETALP, LIG, UJF

385 rue de la Bibliothèque, BP n° 53
38041 Grenoble, cedex 9, France

Najeh.Hajlaoui@imag.fr

Christian Boitet

GETALP, LIG, UJF

385 rue de la Bibliothèque, BP n° 53
38041 Grenoble, cedex 9, France

Christian.Boitet@imag.fr

Abstract

Linguistic porting of content management services processing spontaneous utterances in natural language has become important. In most situations, such utterances are noisy, but are constrained by the situation, thus constituting a restricted *sublangage*. In previous papers, we have presented three methods to port such systems to other languages. In this paper, we study how to also *personalize* them by making them capable of *automatic perception adaptation*, using fuzzy evaluation functions. We have reengineered IMRS, a music retrieval NL-based system, to implement that idea, and ported it to French, English and Arabic using an enhanced version of our *external porting* method, building a *unique* content extractor for these three languages. More than 30 persons participated in a preliminary on-line qualitative evaluation of the system.

1 Introduction

Multilingualizing systems handling content expressed in spontaneous natural language is an important but difficult problem, and very few multilingual services are available today. The choice of a particular multilingualization process depends on the translational situation: types and levels of possible accesses, available resources, and linguistic competences of participants involved in the multilingualization task. Three main strategies are possible in principle for multilingualization, by translation, and by internal or external adaptation. We consider here the

subproblem of *linguistic porting*, where the content is adapted to another language, but not necessarily to a different cultural environment. We also try to add some level of *personalization*, by *automatic perception adaptation*, based on the use of fuzzy evaluation functions. We use the example of IMRS, an *Impression-based Music Retrieval System* (Kumamoto, 2004), with a native interface in Japanese, which we have reengineered and ported to French, English and Arabic. The context and objectives of our work are presented in the second section. The third section presents the IMRS original prototype and the possible strategies to achieve porting and personalization. In the fourth section, we give detailed specifications of our reengineered music retrieval system, IMRS-g. In the fifth section, we present the implementation of five music retrieval modes. Finally, we report on the multilingual porting of this system.

2 Methods for porting NL-based content processing systems

The choice of a method for multilingualizing e-commerce services based on content extraction from spontaneous texts depends on two aspects of the translational situation:

- The level of access to resources of the initial application. Four cases are possible: complete access to the source code, access limited to the internal representation, access limited to the dictionary, and no access. In the case of IMRS, the access was limited to the internal representation, visible as a non-linguistic interface in the original prototype (a set of 10 impressions manipulate by a set of 7 checkbox).
- The linguistic qualification level of the persons involved in the process (level of know-

ledge of the source language, competence in NLP) and the resources (corpora, dictionaries) available for the new language(s), in particular for the *sublanguages* at hand.

We concentrate on NLP-based systems that perform specific tasks in restricted domains. Figure 1 shows the general structure of these systems. Examples of such applications and services are: categorization of various documents such as AFP (Agence France Presse) flash reports or customer messages on an ASS (After Sale Service) server, and information extraction to feed or consult a database (e.g. classified ads, FAQ, automated hotlines).

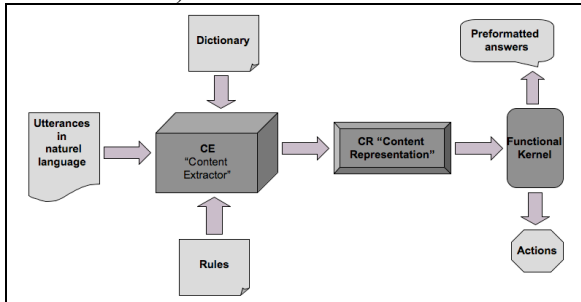


Figure 1: general structure of an NLP-based CMS

We first studied *linguistic porting* of e-commerce systems handling spontaneous utterances in natural languages, that are often noisy, but constrained by the situation, and constitute a more or less restricted *sub-language* (Kittredge, 1982), (Harris, 1968) (Grishman and Kittredge, 1986).

This kind of system uses a specific content representation on which the functional kernel works. In most cases, this content representation is generated from the *native* language L1 by a content extractor. In our PhD, we have identified three possible methods of linguistic porting, and have illustrated them by porting to French CATS (Daoud, 2006), a Classified Ads Transaction System in SMS (Arabic) deployed in Amman on Fastlink, as well as IMRS, mentioned above. The three possible strategies for linguistic porting are internal porting, external porting and porting by machine translation. Figure 2 shows an example of the car domain with the output of the content extractor (CRL-CATS).

In CRL-CATS (Content Representation Language for CATS), a posted SMS is represented as a set of binary relations between objects. It is a kind of semantic graph with a UNL-like syntax (Uchida and Zhu 2005-2006). There are no vari-

ables, but the dictionary is used as a type lattice allowing specialization and generalization.

```

2000 م رينو ميجان م
;Selling Renault Megane m 2000
[S]
sal(saloon:00,sale:00)
mak(saloon:00,RENAULT(country<France,
county<europe):07)
mod(saloon:00,Megane(country<France,
country <europe,make<RENAULT):0C)
yea(saloon:00,2000:0K)
[/S]

```

Figure 2: Example of SMS

2.1 Internal porting

The first possibility consists in adapting the original content extractor of the application from L1 to the target language L2 (see Figure 3); but that is viable only if :

- the developers agree to open their code and tools,
- the code and tools are relatively easy to understand,
- the resources are not too heavy to create (in particular the dictionary).

That method requires of course training the localization team with the tools and methods used.

Under these conditions, adaptation can be done at a very reasonable cost, and further maintenance.

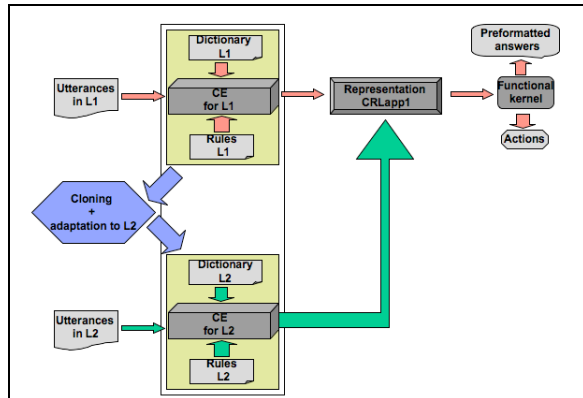


Figure 3: internal porting

We have previously experimented this method (Hajlaoui, 2008) by porting CATS from Arabic to French: for that, we adapted its native Arabic content extractor, written in EnCo¹ (Uchida and Zhu 1999), by translating its dictionary, and modifying a few analysis rules.

¹ EnCo is a tool based on rules and dictionaries used for content extraction in original version of CATS system.

2.2 External porting

If there is access only to the internal content representation, the solution consists in adapting an available content extractor for L2 to the sublanguage at hand, and to compile its results into the original content representation (see Figure 4).

For a company wanting to offer multilingualization services, it would indeed be an ideal situation to have a generic content extractor, and to adapt it to each situation (language, sublanguage, domain, content representation, task, other constraints). However, there is still no known generic content extractor of that power, and not even a generic content extractor for particular languages, so that this approach cannot be considered at present. Our approach is then to adapt an existing content extractor, developed for L2 and a different domain/task, or for another language and the same domain/task.

We also applied this method to port CATS from Arabic to French, and experimentation are described in (Hajlaoui, 2008).

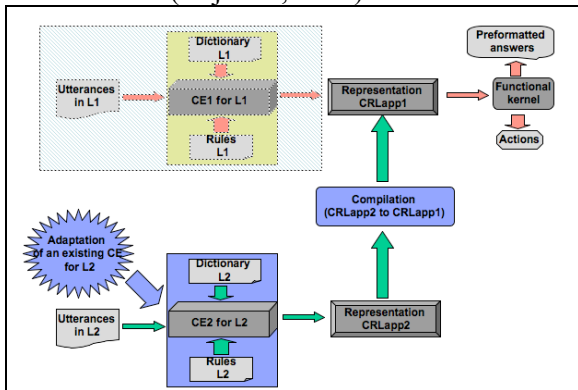


Figure 4: external porting

2.3 Porting by machine translation

If there is no access to the code, dictionary, and internal content representation of the original application, the only possible approach to port it from L1 to L2 is to develop an MT system to automatically translate its (spontaneous) inputs from L2 into L1 (see Figure 5).

Porting CATS from Arabic to French by statistical translation gave a very good performance, and that with a very small training corpus (less than 10 000 words). This proves that, in the case of very small sub-languages, statistical translation may be of sufficient quality, starting from a corpus 100 to 500 smaller than for the general language.

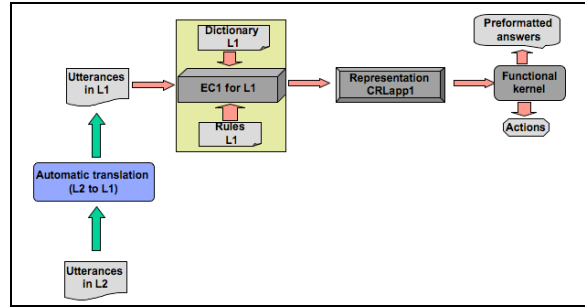


Figure 5: porting by machine translation

2.4 Results and evaluation

We translated manually the evaluation corpus used for the evaluation of CATS Arabic version. It contains 200 real SMS (100 SMS to buy + 100 SMS to sale) posted by real users in Jordan.

We spent 289 mn to translate the 200 Arabic SMS (2082 words is equivalent to 10 words/SMS, approximately 8 standard pages²) into a French translation, or about 35 mn per page, and 10 mn per standard page to pass from raw translation to functional translation.

We obtained 200 French SMS considered to be functional (1361 words, or about 6,8 words/SMS, approximately 5 standard pages). We then computed the recall R, the precision P and the F-measure F for each most important property (action “sale or buy”, “make”, “model”, “year”, “price”).

$$P = \frac{|\text{correct entities identified by the system}|}{|\text{entities identified by the system}|};$$

$$R = \frac{|\text{correct entities identified by the system}|}{|\text{entities identified by the human}|};$$

$$F\text{-measure} = \frac{2PR}{P+R}$$

Table 1 summarizes the percentage (F-measure ratio) of the Arabic-French porting of CATS and shows details in (Hajlaoui, 2008). Properties having numbers as values, like price and year, lower the percentage of porting by external porting, but the advantage is that method requires only accessing the internal representation of the application.

	Minimum	Average	Maximum
Internal porting	95%	98%	100%
External porting	46%	77%	99%
Porting by statistical translation	85%	93%	98%

Table 1: evaluation of three methods used for porting CATS_Cars from Arabic to French.

² Standard page = 250 words

In the third part of this article, we describe the multilinguization of IMRS, IMRS-g, which includes a module of queries management, where the queries are expressed either in a natural language or in a graphical interface showing 10 vectors corresponding to the internal content representation. In response to a query, the user receives a set of music pieces that correspond to her/his desired selection criteria.

In addition to the original design, where the NL expressions of the 10 measures are mapped in a fixed way to the integers in the real interval $[1, 7]$, we have tried to apply a small part of the theory of fuzzy sets to improve the representation and evaluation of human perceptions.

3 Multilinguization of IMRS

To port IMRS to several languages, we used the external porting method and we built a new content extractor, which treats simple utterances related to the music domain.

3.1 IMRS

IMRS (Kumamoto and Ohta, 2003) is a non-deployed Web service prototype, developed as an experimental base for a PhD. It allows to retrieve music pieces either by using Japanese queries, or by manipulating a graphical interface with 10 criteria settable by knobs (speed, noise, rhythm...), and showing remarkable values (integers between 1 and 7) expressed by English labels. In IMRS, an utterance processed by the system is a spontaneous sentence or fragment of a sentence. The content extractor transforms it into a vector of 10 real numbers in the interval $[1, 7]$. The symbol *nil* means *don't care*.

The 10 components are called *Noisy-Quiet*, *Calm-Agitated*, *Bright-Dark*, *Refreshing-Depressing*, *Solemn-Flippant*, *Leisurely-Restricted*, *Pretty-unattractive*, *Happy-Sad*, *Relaxed-Aroused*, *The mind is restored-The mind is vulnerable*. Each has associated grades (interpreted as "concepts" below). For example, the component *Happy-Sad* is characterized by the seven grades: *very happy*, *happy*, *a little happy*, *medium*, *a little sad*, *sad* and *very sad*. In the original IMRS, these values always correspond to the integers in the $[1, 7]$ interval, respectively 7.0, 6.0, 5.0, 4.0, 3.0, 2.0 and 1.0.

A request to find a piece of music that gives a happy impression (*happy*) corresponds to the 10-dimensional vector as follows: (*nil nil nil nil nil nil nil 6.0 nil nil*) (Kumamoto, 2007), but the music pieces can be described by vectors having non-integer components.

Although we had a quite precise description of the internal representation used by IMRS. We could not find information on the rest of the system. Hence, we recreated it to emulate the functions described in the original publications. That includes the system architecture, the design and implementation of the database, the management of requests, and the programming of actually much more than the originally proposed service.

By definition, *linguistic porting* consists in making an application existing in some language L1 available in another language L2, within the same context. Evaluation of the linguistic porting of a content management application can be done at two levels.

- *Evaluation at the internal representation level*. It is an evaluation at the level of components.
- *Evaluation at the task level*. It is an *end-to-end* evaluation of the new version (in L2) of the application.

To make an end-to-end evaluation of IMRS, an IMRS Web-based simulator was developed. It makes it possible to evaluate *in context* the result of linguistic porting (Japanese \rightarrow French, Arabic, English). A real database with real music pieces, characterized by 10-dimensional vectors as in IMRS, was also created.

The aim of the multilinguization was however not to develop an application strictly equivalent to IMRS, with the addition of being able to handle queries expressed in French, English and Arabic, but to develop an upward compatible, extended application. In particular, we wanted to add other dimensions corresponding to the type of music, the composer, the period of composition, the instruments used, etc. We also wanted to experiment the possibility to associate to each impression such as *happy* a fuzzy set over $[1, 7]$ expressed by a membership function (into $[0, 1]$). More details are given below.

3.2 Our IMRS-g system

With the help of a Master student in computer science, Xiang Yin, we have programmed in

PHP/MySQL a Web service called IMRS-g, re-implementing as accurately as possible the system IMRS, and generalizing it.

Not having sufficient expertise in Japanese, we replaced Japanese by French. We also adapted the NLP part to English and Arabic, using the same strategy to handle the three languages.

We then generalized the internal representation by adding other search criteria (such as the type of music, the composer, the period of composition, and the instruments used), and using fuzzy sets.

A large set of music pieces was loaded into the database, and labelled by vectors in a collaborative way. An evaluation of the French version was then conducted as part of a realistic use, with students listening to music.

The first part of the linguistic porting has been very rapid, since it consisted only in translating into French and Arabic the NL labels expressing impressions (*Noisy/Quiet*, *Calm/Agitated*, *Sad/Happy*, etc.), by associating them the same values as in IMRS.

The content extractor processes simple utterances and extracts from them a 10-dimensional IMRS vector, and the additional information in the form (lists of) of items.

As in IMRS, a request for a music piece can be made either by typing a query in natural language, or through a graphical interface allowing to manipulate a 10-dimensional vector, and to fill fields for the other types of information.

In response, the user receives a list of links to music pieces corresponding to its selection criteria. Clicking on a link starts the playing of the corresponding music piece.

3.3 Generalization by *fuzzifying* the interpretation of the NL labels

The original representation of IMRS seems too rigid to express utterances like *quite calm* or to change the current request using an utterance like *a little slower*. Even if we agree that each term corresponds to an interval of length 1 centred on its reference value, e.g. [5.5, 6.5[for *happy*, [6.5, 7.5] for *very happy*, etc., there are problems at the extremities. Therefore we studied the possibility of better modelling and better processing the requests by using fuzzy logic (Zadeh, 1965).

In order to reason from imperfect knowledge, in contrast to classical logic, fuzzy logic pro-

poses to replace the Boolean variables used in classical logic by fuzzy variables, and the classical *crisp* sets by fuzzy sets.

Let U be a universe of elements. A *fuzzy set* A over U is defined by its membership function (f_A). An element x of U is in A with a degree of membership $f_A(x) \in [0, 1]$. If $f_A(x) \in \{0, 1\}$, A reduces to a classic set, where $x \in A$ if $f_A(x)=1$ and $x \notin A$ if $f_A(x)=0$ (f_A is then simply the characteristic function of A).

In a fuzzy set, an element x more or less belongs to the concept associated to A , or to the concept attached to A (such as *happy*). A fuzzy set is defined by all values of its membership function on its definition domain (which may be discrete or continuous).

For example, the concept *young* might be defined over the universe U of possible (integer) ages $U = [0, 120]$ by the discrete fuzzy set $A = ((10 \ 1), (20 \ 0.8), (30 \ 0.6), (40 \ 0.2), (50, 0.1), (60 \ 0), (70 \ 0), (80 \ 0))$. The first pair means that a 10-year old is *100% young*, and the fifth that a 50-year old is *10% young*.

Using fuzzy logic, we could say that a piece of music is *100% rapid* if its tempo is 100 (100 crotchets (quarter notes) per minute), with a bell-shaped curve for the membership function, rising from 0 to 1 and then falling, in the range [84, 112]. Then, *rapid* might be understood as *impression of rapidity*. As the impression of rapidity may differ from person to person, that curve may differ accordingly.

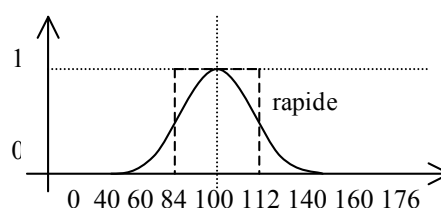


Figure 6. Representation of the rapidity impression

We propose to incorporate the possibility to move from the *perceptions* to digital measurements and to personalize the system by learning parameters of each curve of this type for each user. Such a curve can be characterized by a small number of significant points, such as the maximum, 2 points at the 10% below the maximum, 2 minima on each side, and 2 points at 10% above the global minimum.

To find the criteria for each piece of music, we have developed a website to ask a group of people to listen to music pieces and to give their opinions in terms of impressions, knowing that they will not have the same taste and the same perception. For example, for the same piece, a first listener will say that it is rapid, and a second will find it very rapid. The question here is how to merge these different views into a single impression. We propose two solutions: (1) construct a fuzzy set which is the *average* of those of annotators, possibly by giving greater weight to the annotations of *confirmed* annotators, (2) build several *perception types*, i.e. several fuzzy sets corresponding to subgroups of users with similar perceptions. We know that the Japanese persons find only slow pieces of music that Westerners find very slow.

In this work, we have taken into account previous queries of users or the history of users. For example, if a user requests a piece *a little less noisy*, or *a little more calm*, we should be able to use the information saved in his history, and calculate the new request taking into account the perceptions associated to the last piece of music listened to.

4 Specification and implementation

We specified and implemented a multimedia database for storing music pieces, as well as information representing the impressions of each piece. As said above, we added to the 10 features of IMRS other information types: singer, poet, composer, genre, album and duration, for each music piece. Moreover, to evaluate music, we stored the values of the impressions recorded by contributing users for each piece. These values were used to produce the final values stored in the database. To analyze the impressions of users, we requested further information from each user, as gender and age.

We loaded our database with a set of 354 pieces (89 Western, 265 Eastern) and all information related to each piece (artist, album, genre...). The duration of individual pieces varies between 48 seconds and 22 minutes.

The website has a login page that allows a secured access for each user. For a first connection, the user must register and fill some information from which we compute and store a profile.

If the connection is successful, a list of pieces is displayed. For each piece, a link allows listening to the music and also opens a new page providing an adapted evaluation interface appropriate to the evaluation task.

In the evaluation phase, the user can listen to the selected piece and evaluate it according to the 10 IMRS basic criteria (soft, calm, happy...). For each criterion, we offer a range of values and the user can move a cursor and put it on the value that represents its perception. Next, we propose several ways to search for music pieces.

The cost of multilinguization of the IMRS system was 3 man-months. To this cost, we add 1 man-month for the development and integration task of the content extractor for the three languages (French, Arabic, English).

5 Music retrieval modes

After registering and connecting, users listen to and evaluate music. The evaluation information is recorded directly in the database.

For each dimension, we compute the average of the obtained values. This phase is temporary pending further evaluations to draw the curves associated to each dimension and to each piece.

We defined and implemented five possibilities to search music: by user profile, by impressions, by selecting criteria, by input utterances, and by history.

5.1 Search by user profile

We propose to users music adapted to their preferences recorded in their profiles. The method follows the following steps:

- Find the user profile (ten values that represent his basic impressions) in the database.
- Compute the Euclidean distance between the two vectors formed by the 10 values of profile and the 10 values of each music piece (see example below).
- Sort pieces by distances in ascending order.
- View the nearest 10 pieces.

Here is an example:

User profile: impressions vector

Profile = (Nil 6 3 Nil 2 1 3 5 Nil Nil)

Piece impressions (existing impressions vector):

Piece1 =

(3.5 Nil 2.3 5.0 3.2 Nil 2.6 Nil 6.0 1.4)

Euclidian distance (d):

$$d = \sqrt{(4-3.5)^2 + (6-4)^2 + (3-2.3)^2 + (4-5)^2 + (2-3.2)^2 + (1-4)^2 + (3-2.6)^2 + (5-4)^2 + (4-6)^2 + (4-1.4)^2}$$

=> d = 5.3,

Note: if value = "Nil", we put value := 4.

5.2 Search by impressions

We ask the user to place the cursors on the values that represent his perception. We can limit ourselves to a particular type of music (Western music, Eastern music or light music). The search method has the following steps:

- Choose the kind of music (Western music, Eastern music or light music).
- Place one or more cursors on the values that represent user's perception.
- Compute the Euclidean distance between the two vectors formed by the 10 values of search and the 10 values of each piece.
- Sort pieces by distances in ascending order.
- View the nearest 10 pieces.

Here is an example: we search a noisy (*fort*) and somewhat calm (*assez calme*) piece (see Figure 7).

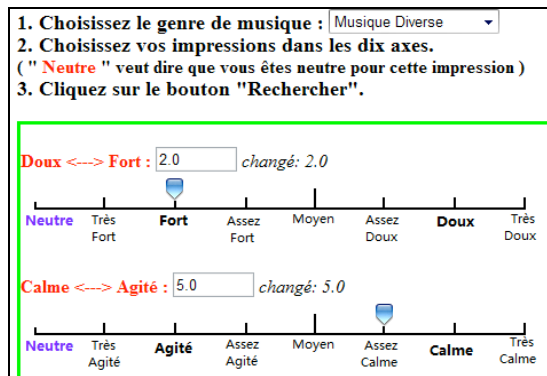


Figure 7. Example of search by impressions

The result of the previous request is a set of 10 music pieces.

5.3 Search by selection criteria

We offer four search criteria: artist, album, genre and creation date. The search methods for each of these criteria are similar.

For example, the search by artist follows the following steps:

- Search all artists (singers) existing in the database.
- Choose an artist from this list.
- Search all pieces performed by this artist and show them (by links).

5.4 Search by input utterances

The content extractor works for French, Arabic and English, and handles simple utterances related to the music domain. This program takes as input a corpus of music pieces and gives as output a file containing the corresponding vector representations.

The search method has the following steps:

- Enter an utterance in natural language representing impressions of music search.
- Call a content extractor. The result, which contains a vector representing the desired perceptions, is stored in a text file.
- Extract the vector from the text file.
- For each value of the vector (V_v), if one of the symbols (+, ++, -, --, -) appears, then we extract the value of the last search of the concerned user (V_o : old value) from the database to compute the new value of search (V_n : new value).

Here are some examples of utterances that correspond to the precedent symbols. +: more noisy, ++: still more noisy, -: less noisy, --: still less noisy, -: not noisy.

We treat these symbols with the following rules:

$$\text{If } (V_v == '+') \quad \{V_n = V_o + \alpha ;\}$$

$$\text{If } (V_v == '++') \quad \{V_n = V_o + 2\alpha ;\}$$

$$\text{If } (V_v == '-') \quad \{V_n = V_o - \alpha ;\}$$

$$\text{If } (V_v == '--') \quad \{V_n = V_o - 2\alpha ;\}$$

$$\text{If } (V_v == '-x') \quad \{V_n = 7 - x ;\}$$

$$\text{If } (V_n > 7) \quad \{V_n = 7 ;\}$$

- Compute the Euclidean distance between the two vectors formed by the 10 desired values and the 10 values of each piece.
- Sort music by distances in ascending order.
- View the nearest 10 pieces.

5.5 Search by history

We extract from the history of each user five types of information: (a) the kind of desired pieces, (b) their creation date, (c) the artists (performers), (d) the liked albums, (e) the favourite impressions.

The search method has the following steps:

- Search the user's history in the database and check if the user has already searched with the five previous conditions.
- If the user has searched for condition (a) or (b) or (c) or (d), we extract the last value of found for each of them. 4 values are obtained.

If the user searches by impressions (condition (e)), we compute for each dimension the average that represents the history of the searches.

- Search for music using the values obtained at step 2.

If (e) is verified, we compute the Euclidean distance between the average of impressions representing the history and impressions existing in the database.

If (e) is not verified, we look for pieces, using only the 4 values obtained by the conditions ((a), (b), (c), (d)).

Here an example of a history of one user. For condition (a), the latest search value is “Pop”. For condition (b), there is no value, i.e. the user did not search by creation date. For condition (c), the latest search value is “1” (number of the artist). For condition (d), the latest search value is “2” (number of the album). For condition (e), there are 3 vectors in the search history:

V1=(2 5 Nil 3 Nil 2 7 1 Nil Nil)
V2=(3 Nil 4.5 2.5 Nil 3.1 6.4 Nil 5 2)
V3=(3.5 4.3 Nil 2.1 Nil Nil Nil 3 Nil Nil)

We compute the average of the history, V_m :
 $V_m=(2.83 4.65 4.5 2.53 Nil 2.55 6.7 2 5 2)$

We search for pieces that verify the complex condition: (Kind of music = 'Pop') AND (Number of the artist = '1') AND (Album ID = '2') AND (Impressions vector is closest to V_m according to the Euclidean distance).

If the search is successful, then the result is optimal. Otherwise, we search pieces that correspond to the second condition: ((Kind of music = 'Pop') OR (Number of the artist = '1') OR (Album ID = '2')) AND (Impressions vector is closest to V_m according to the Euclidean distance).

We refined this search through other combinations formed by the conditions (a), (b), (c), (d) and (e) and differentiated by the OR and AND operators.

6 Multilingual porting

To build our content extractor, we started from a content extractor for French we had previously develop for the same domain, integrated it into IMRS-g, and extended it as explained above (more information type, and fuzzy sets). We then ported it to English and to Arabic, using the sec-

ond technique of external porting (when one has access to the internal representation).

Seeing the large percentage of common code to the 3 content extractors obtained, we factorised it and obtained a unique content extractor handling input utterances in the music domain in our 3 target languages: French, English and Arabic. This technique is perhaps not generalizable, but it works for this sub-language, which is very small, and for the simple task of extracting information representable in very small textual fragments.

Here are some examples of results for Arabic, French and English:

Exemple_Ar 1 : أريد قطعة موسيقية جد هادئة //je veux un morceau de musique très calme
Musique_Ar 1 : musique-spec=(nil 7,0 nil nil nil nil nil nil nil nil)
Exemple_Ar 2 : أريد قطعة موسيقية قليلة الضجيج //je veux un morceau de musique un peu bruité
Musique_Ar 2 : musique-spec=(3,0 nil nil nil nil nil nil nil nil nil)
Exemple_Fr 1 :je veux un morceau de musique calme et très solennel
Musique_Fr 1 : musique-spec=(nil 6,0 nil nil 7,0 nil nil nil nil nil)
Exemple_Fr 2 :je veux un morceau de musique assez fort et clair
Musique_Fr 2 : musique-spec=(3,0 nil nil 6,0 nil nil nil nil nil nil)
Exemple_En 1 :I want a calm and very solemn music
Musique_En 1 : music-spec=(nil 6,0 nil nil 7,0 nil nil nil nil nil)
Exemple_En 2 :I want a little noisy and bright music
Musique_En 2 : music-spec=(3,0 nil nil 6,0 nil nil nil nil nil nil)

Tableau 1: Examples of results of IMRS-g for Arabic, French and English

Conclusion

We have presented several possible methods for "porting" applications based on handling the content of spontaneous NL messages in a "native" language L1 into another language, L2. In a previous paper, we described experiments and evaluations of these methods.

We tried to do an “end-to-end” evaluation of porting IMRS by building a website that proposes to engage people in evaluation of a set of music pieces, thereby offering them to search for

music in different possible modes. To that effect, we have produced a functional Web site

(<http://www-clips.imag.fr/geta/User/najeh.hajlaoui/Musique/>). To date, the evaluation has been done only for French. More than 30 users have participated, perhaps because they were rewarded in a sense: as a kind of compensation, each user could listen to appropriate music adapted to his way of perception and taste. The use of fuzzy logic proved useful and was perhaps even necessary to give some freedom of choice of impressions to users.

Acknowledgments

We thank Yin Xiang, who dedicated his TER internship during his master in CS to the study and implementation of the IMRS-g system.

We would also like to thank our reviewers, who made many constructive remarks and suggestions, which we gladly incorporated in this new version of the paper.

References

- Daoud, D. M. (2006). *It is necessary and possible to build (multilingual) NL-based restricted e-commerce systems with mixed sublanguage and contend-oriented methods*. Thèse. Université Joseph Fourier. Grenoble, France. September 23, 2006. 296 p.
- Grishman, R. and R. Kittredge. (1986). *Analyzing language in restricted domains*. Hillsdale NJ. Lawrence Erlbaum Associates. 248 p.
- Hajlaoui, N. (2008) *Multilinguisation de systèmes de e-commerce traitant des énoncés spontanés en langue naturelle*. Thèse. Université Joseph Fourier. Grenoble. 25 septembre 2008. 318 p.
- Harris, Z. (1968). *Mathematical structures of language*. in *The Mathematical Gazette*. Vol. 54(388): pp. 173-174. May, 1970.
- Kittredge, R. (1978). *Textual cohesion within sublanguages: implications for automatic analysis and synthesis*. Proc. Coling-78. Bergen, Norvège. August 14-18, 1978. Vol. 1/1.
- Kittredge, R. (1982). *Variation and Homogeneity of Sublanguages*. in *Sublanguage - Studies of Language in Restricted Semantic Domains*. Walter de Gruyter. Berlin / New York. 20 p.
- Kittredge, R. (1993). *Sublanguage Analysis for Natural Language Processing*. Proc. First Symposium on Natural Language Processing. Thailand, Bangkok pp. 69-83.
- Kittredge, R. and J. Lehrberger (1982a). *Sublanguage - Studies of language in restricted semantic domain*. Walter de Gruyter. Berlin / New York.
- Kumamoto, T. (2004). *Design and Implementation of Natural Language Interface for Impression-based Music-Retrieval Systems*. Knowledge-Based Intelligent Information and Engineering Systems. Springer Berlin / Heidelberg. October 14, 2004. Vol. 3214/2004: pp. 139-147.
- Kumamoto, T. (2007). *A Natural Language Dialogue System for Impression-based Music-Retrieval*. Proc. CICLING-07 (Computational Linguistics and Intelligent Text Processing). Mexico. February 12-24, 2007. 12 p.
- Kumamoto, T. and K. Ohta (2003). *Design and Development of Natural Language Interface for an Impression-based Music Retrieval System*. in *Joho Shori Gakkai Kenkyu Hokoku*. Vol. 4(NL-153): pp. 97-104.
- Kurohashi, S. and M. Nagao (1999) *Manual for Japanese Morphological Analysis System JUMAN*. Rap. Language Media Lab. School of Informatics, Kyoto University. Kyoto, Japan. November 1999.
- Uchida, H., M. Zhu, et al. (2005-2006). *Universal Networking Language* 10 2-8399-0128-5. 218 p.
- Uchida, H. and M. Zhu (1999). *Enconverter Specifications*, UNU/IAS UNL Center, 33 p.
- Zadeh, L. A. (1965). *Fuzzy sets*. *Information and Control* 8: pp. 338-353.