Internal Use Only（非公開）

TR-SLT-0056

Towards Multilingual Translation of the BTEC Aided by Analogy

類推に基づくBTECコーパスの多言語化への検討

ギエーム・ペラルタ　　イヴ・ルパージュ
Guilhem Peralta　　Yves Lepage

December 15, 2003

We show how analogy relations can structure a corpus (here, the BTEC) into paradigmatic tables displayed in a human-readable manner in an HTML interface. We also show that such a structure enables the generation of new sentences the quality of which may be checked by the user using the HTML interface. Moreover, this technique applied in parallel on a bicorpus is a way of implementing a method of machine-aided translation.

（株）国際電気通信基礎技術研究所
音声言語コミュニケーション研究所
〒619－0288「けいはんな学研都市」光台二丁目2番地2 TEL：0774－95－1301

Advanced Telecommunication Research Institute International
Spoken Language Translation Research Laboratories
2-2-2 Hikaridai "Keihanna Science City" 619-0288,Japan
Telephone:+81-774-95-1301
Fax　:+81-774-95-1308

Contents

1 Introduction

In this report, we show how the properties of the analogy relation applied to a large aligned multilingual corpus have allowed us to extract information from it, and more precisely to organize its content. Two different goals were made possible in this way: creating new utterances and translating.

1.1 The data: BTEC

The objective of the ATR SLT Laboratory is speech-to-speech translation. This theme is applied to a real-life problem: the difficulties a traveller may face when visiting a country whose language he hasn't mastered.

The global task has been divided into three parts, each delegated to a department: voice recognition, text translation, and voice synthesis.

I worked in department 3, whose task is the translation of such utterances. The data used as a knowledge base is the Basic Traveller's Expressions Corpus, BTEC for short[CSTAR 03]. It is an aligned multilingual corpus, whose elements are simple sentences that a traveller may utter when going to another country. For instance, *I'd like to book a room.*, *Please call a taxi for me.*, etc.

Today, there are about 160,000 sentences translated into Japanese, Chinese, Korean, English... and new ones are being added regularly, as well as new languages (Italian, ...)

1.2 The structuring relation: Analogy

All of the work presented in this report relies on analogy. This simple relation has been proved to be of significant importance in linguistics. Analogy is a general and intuitive concept, equivalent to the idea of proportion. It links four objects A, B, C and D together, where A is to B what C is to D. In this research, we focus on the case of strings of symbols.

We use a purely formal characterisation of analogy between strings of symbols that is based on the verification of a similarity criterion citeitk:ana:

$$A : B = C : D \quad \Rightarrow \quad \begin{cases} \mathrm{dist}(A,B) & = & \mathrm{dist}(C,D) \\ \mathrm{dist}(A,C) & = & \mathrm{dist}(B,D) \end{cases}$$

where dist(A, B) stands for the edit distance between A and B with deletions and insertions as the sole edit operations, and of a contiguity criterion:

$$A : B = C : D \quad \Rightarrow \quad \forall a, |A|_a + |D|_a = |B|_a + |C|_a$$

,

where $|A|_a$ is the number of occurrences of the symbol a in the string A.

For instance, for the analogy

I prefer Italian food. : *I prefer Japanese food.* = *I like Italian food.* : *I like Japanese food.*, the following holds:

$$\begin{cases} \mathrm{dist}(\textit{I prefer Italian food., I prefer Japanese food.}) & = & 9 \\ \mathrm{dist}(\textit{I like Italian food., I like Japanese food.}) & = & 9 \end{cases}$$
$$\begin{cases} \mathrm{dist}(\textit{I prefer Italian food., I like Italian food.}) & = & 8 \\ \mathrm{dist}(\textit{I prefer Japanese food., I like Japanese food.}) & = & 8 \end{cases}$$
$$\begin{cases} |\textit{I prefer Italian food.}|_e & + & |\textit{I like Japanese food.}|_e & = & 2+3 & = & 5 \\ |\textit{I prefer Japanese food.}|_e & + & |\textit{I like Italian food.}|_e & = & 4+1 & = & 5 \end{cases}$$

and similarly for each letter.

In the sequel of this document, the term "analogy" refers to the relation defined as above[Lepage 01].

1.3 The structured data: Paradigm tables

Paradigms are another linguistical concept, related to morphology and syntax. They are better illustrated by morphology. They concern only one language at a time. As it is very intuitive, let's look at an example in the form of a paradigmatic table:

chanter	valider	travailler
je chante	je valide	je travaille
tu chantes	tu valides	tu travailles
il chante	il valide	il travaille

This table reveals the regularity of the process of creating these utterances. This phenomenon is very often present in linguistics, and is what enables us to understand words we have never heard before. In the same way, somebody who learns a new regular English verb does not have to learn its past form, he just has to know that it is regular.

Hence, nothing is lost when extracting the following data from the above table:

chanter	valider	travailler
je chante		
tu chantes		
il chante		

Anyone will be able to automatically fill in the holes if he knows that *valider* is conjugated the same way *chanter* is. And anyone who knows *valider* will understand *je valide* even if he has never heard it before.

The way that this is achieved is by solving analogical equations: every single verb form in the table can be deduced from the first row and the first column. There is enough information in (*chanter, travailler, tu chantes*) to produce *tu travailles*, as it is the solution of the following analogical equation: *chanter* : *travailler* = *je chante* : x.

A paradigmatic table may be abstracted in the following manner:

A	B_1	B_2	$\cdots$	B_n
C_1	D_{11}	D_{12}	$\cdots$	D_{1n}
C_2	D_{21}	D_{22}	$\cdots$	D_{2n}
C_3	D_{31}	D_{32}	$\cdots$	D_{3n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
C_m	D_{m1}	D_{m2}	$\cdots$	D_{mn}

where $\forall(i,j),\ A : B_i = C_j : D_{ij}$, and where any cell can play the role of A because there is no fundamental difference between rows[1].

We can extend this idea to more complex objects[Itkonen & Haukioja 97], such as, for instance, the sentences of our BTEC. These commutations can be done at various levels:

[1] This assumes that analogy is a transitive relation, which has been proved to be wrong. However, the cases of non-transitivity are so rare, and especially so when dealing with real-life utterances, that we chose to make this hypothesis.

they can just as well be the change of a letter or the replacement of any substring[2] by any other (another word, a proposition, nothing, ...).

The BTEC is a corpus well fitted to this research. It contains many sentences that look a lot alike: for instance, *I like Japanese food.*, *I like Italian food.*, *I feel like seafood.*, *How about some Italian food.*, ... which made us believe that we could obtain relatively large tables. As, of course, the corpus does not contain all the utterances possible, there will be empty cells in the paradigms we might extract, but if we focus on a specific domain that is well covered by the corpus, the paradigm should be quite full[3]. So with an appropriate algorithm, we may be able to generate paradigmatic tables filled with BTEC sentences. A first application of such tables is to expand the corpus by filling in the holes in the tables. A second application is to align different languages to perform translation. We shall illustrate the two applications in the remainder of this report.

[2] A substring is not necessarily connex.

[3] It does, however, rely heavily on the kind of sentences present in the corpus.

2 Paradigm Tables

2.1 Construction

(2.1.1) A symmetrical representation of a symmetrical concept

Our objective thus consists in finding paradigms in the BTEC. This can give linguistic indications of the content of the corpus by clearly exposing certain morphological phenomena.

In tables such as the one presented in the previous section, no particular order exists among rows and columns. In the paradigm itself, no cell plays a different role from the others. *chanter* was chosen as an axis above, but any cell could have been chosen instead.

In order to build paradigms, we have to ask the following question: what characterizes paradigms? The choice we made was to build tables by starting from particular sentences. Those particular sentences will be placed at the top left corner in paradigm tables; they will be named *seed sentences* and they will be noted A. An important point is that A may or may not be in the corpus.

In fact, an analogy can be represented in the following way:

My monkey likes bananas.	My monkey lives in the forest.
I like bananas.	I live in the forest.

which can be seen as the simplest paradigm possible, with A = *My monkey likes bananas.*

Any analogy involving A is such a square. We want to represent any such analogy in a unique table, which will contain all the elements of the corpus sharing an analogical relation with A[4].

The above relation is symmetrical. In the above example, if we fix A, the three other ones separate into two groups: B (*My monkey lives in the forest.*) and C (*I like bananas.*) on one hand, and D (*I live in the forest.*) on the other hand. Hence our algorithms have to handle B and C in the same way; this can be represented by the symmetry of the following table:

My monkey likes bananas.	My monkey lives in the forest.	I like bananas.
My monkey lives in the forest.		I live in the forest.
I like bananas.	I live in the forest.	

So the extended table obtained with the fusion of all these analogies will also be symmetrical.

(2.1.2) Searching for analogies

A naive idea would consist in searching for all analogies involving A and any three sentences of the corpus. The problem is that due to the fact that this relation involves four objects, the complexity of such a search is $O(n^3)$. This would require an unreasonable amount of time.

One solution would consist in computing in advance all the analogies contained in the corpus, to make any further search linear; this solution did not satisfy us however, because in the case in which A is not in the corpus[5], it would have been of no use. And moreover, we cannot instantaneously extract information from a new corpus this way[6].

[4] Except for analogies such as $A : A = B : B$, which would fill the table with uninteresting data.

[5] This will happen in any translation application such as the one presented in Section 4.

[6] With our algorithm, the search for all analogies on the English BTEC (90,000 lines) took three weeks on a biprocessor Pentium IV 2.8GHz machine with a 2Gb memory.

The approach we chose was to reduce the search space. This can be done either by only searching among sentences containing a certain string (related to A), or by only keeping the utterances nearest in some way[7]. We are in fact searching for a context.

In our experiments, we have chosen to filter (using a simple `grep` command) the corpus by a string given by the user. Tests showed that with the machine configuration we used[8], the computation time is reasonable (about a second) with subcorpora of up to 5000 sentences.

(2.1.3) Building the table

In the first step, we will simply organize all the analogies involving A. So the first objective consists of searching for all the (B,C,D) in the subcorpus such that $A : B = C : D$. This single task required much effort to reduce the computation time.

From all these (B,C,D) triples of sentences, we then make two sets: the $\mathcal{BC}$, whose elements will be located on the first row and column of our table; and the $\mathcal{D}$, whose elements will be within the table.

For instance, for the seed sentence *I like Japanese food.*, if the four sentences of the following analogy:

I like Japanese food. : *I like Italian food.* = *I prefer Japanese food.* : *I prefer Italian food.* are present in the corpus, *I like Italian food.* and *I prefer Japanese food.* will be added to $\mathcal{BC}$, whereas *I prefer Italian food.* will be added to the $\mathcal{D}$ set.

Then, all the sentences in $\mathcal{BC}$ are put in the first column and the first row of the table, in a symmetrical manner; all analogies found can then be represented inside the table (and in fact twice because of the symmetry of the table).

Fig. 1 is an example of what we obtained with the seed sentence *I prefer Italian food.* and the filter *food.* It is just an extract of a result we obtained. The actual table is 18x18 but could not fit onto the page.

I prefer Italian food.	*I prefer Japanese food.*	*I prefer French food.*	*I like Italian food.*	*My favorite food is Italian.*	*I prefer seafood.*	*I'd like to have some Italian food.*
I prefer Japanese food.			*I like Japanese food.*			*I'd like to have some Japanese food.*
I prefer French food.						*I'd like to have some French food.*
I like Italian food.	*I like Japanese food.*				*I like seafood.*	
My favorite food is Italian.						
I prefer seafood.			*I like seafood.*			
I'd like to have some Italian food.	*I'd like to have some Japanese food.*	*I'd like to have some French food.*				

Figure 1: A table with all the analogies involving *I prefer Italian food.* $\mathcal{BC}$ is in the first row and column, and the elements of $\mathcal{D}$ are within the table.

[7] Both in terms of distance and similarity, interesting results are obtained.

[8] Equipped with an Intel Pentium III 600 MHz processor.

The next step is again based on a simple idea: some cells cannot be filled, because some analogical equations have no solution. For instance,

I prefer Japanese food. : *I prefer Italian food.* = *I prefer French food.* : x

the contiguity criterion would require that the letter J appears "minus one times" in x, which is not possible[9]. There are such cells in the symmetric table. As a trivial example, for any B which contains a letter absent from A, the equation $A : B = B : x$ has no solution. That situation being almost always true, this explains why the diagonals of many symmetrical matrices are almost empty.

Hence we search for all cells that do not meet the analogy criteria, and we represent them with a •. Fig. 2 shows this for the same table as Fig. 1. A structure appears quite clearly, that will be exploited in the next section. We shall call cells with a • black cells.

I prefer Italian food.	*I prefer Japanese food.*	*I prefer French food.*	*I like Italian food.*	*My favorite food is Italian.*	*I prefer seafood.*	*I'd like to have some Italian food.*
I prefer Japanese food.	•	•	*I like Japanese food.*		•	*I'd like to have some Japanese food.*
I prefer French food.	•	•			•	*I'd like to have some French food.*
I like Italian food.	*I like Japanese food.*		•	•	*I like seafood.*	•
My favorite food is Italian.			•	•	•	•
I prefer seafood.	•	•	*I like seafood.*	•	•	
I'd like to have some Italian food.	*I'd like to have some Japanese food.*	*I'd like to have some French food.*	•	•		•

Figure 2: Same table as Fig. 1, but with analogical impossibilities represented.

2.2 Making tables more significant

(2.2.1) Actual paradigmatic tables are not symmetrical

The table of Fig. 2 is too big, and should be arranged for a better view: in the intuitive conception of a paradigm, there is no symmetry. A paradigm opposes two series of commutation: in our introduction example, one consists in commuting verb roots, and the other one in commuting persons and tenses for a same verb. Applied to our first example, the data contained in the symmetrical table should look this way (after grouping together similar lines):

chanter	valider	travailler	je chante	tu chantes	il chante
valider	•	•	je valide	tu valides	il valide
travailler	•	•	je travaille	tu travailles	il travaille
je chante	je valide	je travaille	•	•	•
tu chantes	tu valides	tu travailles	•	•	•
il chante	il valide	il travaille	•	•	•

[9] Unless we define a special type of string where letter counts could be negative.

This kind of table lacks some structural information, because it just displays all possible analogies without structuring them according to natural and meaningful axes. In the above example, it is obvious that there are two types of elements in $\mathcal{BC}$: *valider* and *travailler* on one hand, and *je chante*, *tu chantes* and *il chante* on the other, hence making a separation obvious. We looked for an algorithm that performs this kind of cut.

(2.2.2) Why we can and should cut the $\mathcal{BC}$ set into two parts

We come from the hypothesis that $\mathcal{BC}$ is made of two commutation series, as in Section (2.2.1). Those two series are what we will try to oppose in a compact table. Such an assumption considerably simplifies the problem but may not always be justified. A paradigm table is not necessarily limited to two dimensions. For instance, the verb morphology in French follows many directions: the verb itself, the person, the time, the form (affirmative / interrogative / passive), etc. But as we could not easily represent a non planar table, we will stay with two dimensions.

We now want to cut the $\mathcal{BC}$ into two parts (a $\mathcal{B}$ and a $\mathcal{C}$, the order does not matter) to be opposed. How to do so? What criteria should the best cut meet? A few ideas came out naturally:

- two lines that share the same behaviour (*i.e.*, having cells with a • located at the same place) might correspond to the same kind of morphological transformation, and hence should be put together.

- a compact table is more reliable if it contains as few cells with a • as possible. Ideally, there is no such cell in a compact table, but as there are more than two axes of commutation, such a situation may occur.

- a compact table is more reliable if it contains as many attested cells as possible, because it means we managed to oppose sentences that had to be opposed.

We could of course compute all the compact tables corresponding to the 2^{n-1} (where n is the cardinal of $\mathcal{BC}$) cuts possible, and choose the best one according to our criteria.

This would not cause any problem for small matrices, but as the complexity is exponential, a better idea had to be found[10].

(2.2.3) How we cut the $\mathcal{BC}$ set into two parts

The observation of simple cases is often of great help, as it was here. When two commutation series appear clearly, we obtain this kind of symmetrical table (after grouping similar rows/columns together):

A	BC_1	BC_2	BC_3	BC_4	BC_5	BC_6
BC_1	•	•	•	•		
BC_2	•	•	•	•		
BC_3	•	•	•	•		
BC_4	•	•	•	•		
BC_5					•	•
BC_6					•	•

[10] It may be in fact feasible but this path has not been explored.

Here, there are two types of lines, one being the negative of the other. One of our initial ideas was that if the rows corresponding to two sentences from $\mathcal{BC}$ are the same, they should be put together in the same set; in the same way, if those two rows are the negative of one another, this means the two sentences must be opposed (*i.e.*, put in different sets).

We can employ this simple case to process more complex and less beautiful symmetrical matrices. There are sentences we definitely want to be opposed; others we want to be grouped together. So the following measure has been introduced between rows of the table, to characterize this "affinity". A vector is associated to each row; its coefficients are 0 if the corresponding cell is a black cell, 1 otherwise. The distance between two rows is the number of different corresponding coefficients, which amounts to computing the Hamming distance. For instance, $dist(BC_1, BC_2) = 0$ and $dist(BC_3, BC_5) = 6$ on the previous table.

So the distance between two lines is minimal if their behaviour is the same, and maximal if they are the negative of each other. And we would like the sentences of a same set ($\mathcal{B}$ or $\mathcal{C}$) to be as near as possible from one another.

Considering the affinity of every element of $\mathcal{BC}$ with any other would not be manageable. So the problem was linearised:

- a best sort is found (according to the criteria described below)
- this sort is accepted as the concatenation of the new $\mathcal{B}$ and $\mathcal{C}$ sets; we search for the best place to cut it into two parts.

More precisely, our compact table extraction algorithm is as follows:

- compute all distances between rows of the table
- take the first row (noted r_0) of a couple which maximises this function
- sort the rows according to their distance to r_0, while grouping together those whose black cells are located in the same place.
- once this sort is achieved, choose the cut that maximises the following criteria:
  - a better compact table contains fewer black cells
  - a better compact table contains more attested cells.

(2.2.4) Example and things to improve

The previous algorithm applied to the table in Fig. 2 returns the table in Fig. 3.

I prefer Italian food.	*I prefer Japanese food.*	*I prefer French food.*	*I prefer seafood.*
I like Italian food.	*I like Japanese food.*		*I like seafood.*
My favorite food is Italian.			●
I'd like to have some Italian food.	*I'd like to have some Japanese food.*	*I'd like to have some French food.*	

Figure 3: The computed paradigm associated with the seed sentence *I prefer Italian food.*

Few phenomena make the task complex, which results in tables of average quality. First, there may be alignments in characters which do not represent any actual morphological phenomenon: our definition of analogy remains at the character level, hence it does not seem to reflect the utterance production process as performed by human beings. Characters are characteristic of the written language; but the written language is just a projection of the spoken one, and has its own rules which will interfere.

However, our experiments showed that the use of a low-level relation that makes no high-level assumption may lead to the emergence of complex and high-level phenomena. The morphology relative to this operation can correspond to any level: it can correspond to orthographic rules as well as conjuguation or higher linguistic objects. This is because a sentence is only seen as a string of symbols; for instance, the space character is considered in the same way as any other. The concept of word is forgotten here[11], mainly because in languages such as Japanese or Chinese, their definition is not that clear. But the choice we made can be discussed, and is still an open problem: at which level should we compute analogies? Is there a coding that is better than any other?

2.3 Interface

(2.3.1) Command line

The program that implements the task described earlier is named `extork`. A typical use would be:

```
cat-btec en | grep "food" | extork -A="I like Japanese food."
```

The `cat-btec` command extracts the sentences from the BTEC (here the English ones); we then filter this stream with `grep` to keep only those sentences that contain the key-word *food.*

The subcorpus obtained is the input of `extork`, which displays the best compact table it computed (according to the algorithm described in Section 2.2) on the standard output. The user can also directly view the large symmetrical table sorted or unsorted.

(2.3.2) HTML and JavaScript

Text-only visualisation is not suited to our kind of data, because tables may get quite large very quickly. Thus, it became obvious that some graphical interface is necessary. In order not to reinvent the wheel, the interface chosen is the couple formed by HTML and JavaScript, which offers all the functinnalities we needed. High-level functions are available and easy to use.

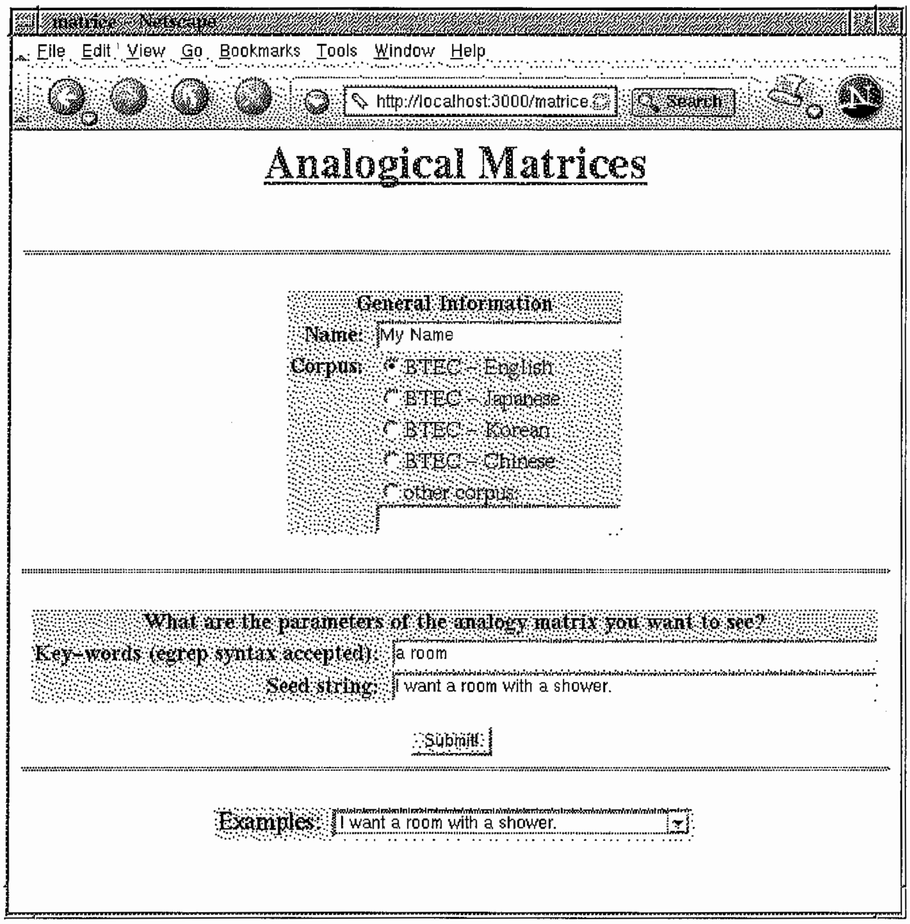
The designed interface serves as a frontend to `extork`. It does, of course, mainly use this program, but it offers other interesting features.

An example can be seen in Fig. 4.

Many fields are self explanatory. The user must enter:

- the name of the session. This will label the search and the eventual work of the user.
- the corpus in which to search. The user may choose any language among those available in the BTEC, or any custom corpus present on the server. A custom corpus is just a file containing sentences separated by carriage returns.
- the keyword with which the corpus may be filtered. The sentences in the table will be searched for in the corresponding subcorpus.
- the seed sentence: this is the utterance from which the table is computed.

[11] We also conducted tests with words and not characters as the symbol units. The resulting tables are smaller, which shows that less phenomena are captured than with characters as symbol units.


matrice - Netscape
File Edit View Go Bookmarks Tools Window Help
http://localhost:3000/matrice
Search

Analogical Matrices

General Information

Name: My Name

Corpus: ● BTEC - English
○ BTEC - Japanese
○ BTEC - Korean
○ BTEC - Chinese
○ other corpus:

What are the parameters of the analogy matrix you want to see?

Key-words (egrep syntax accepted): a room

Seed string: I want a room with a shower.

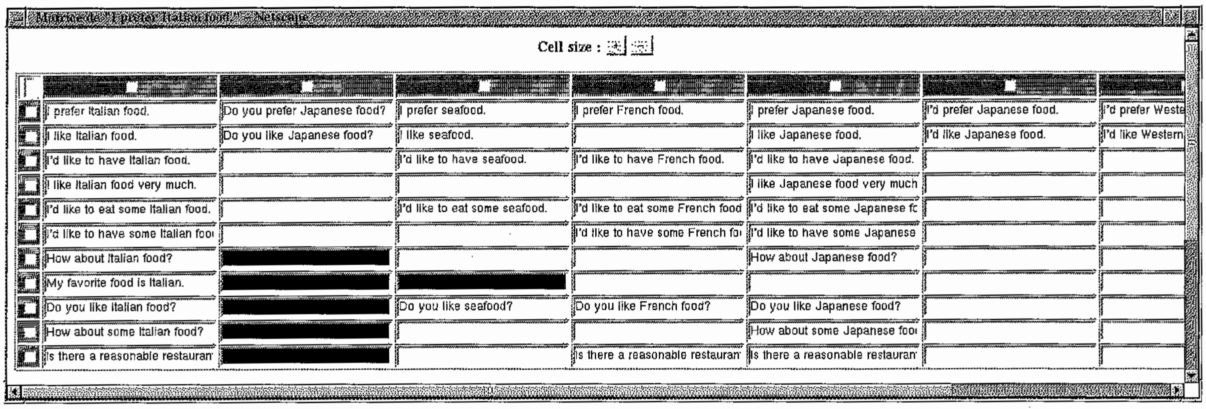
Submit!

Examples: I want a room with a shower.

Figure 4: The main page of the HTML interface.

- a list of examples are available; by selecting them, the corresponding fields are automatically completed and the search can be launched.

The validation of this form calls up a CGI program that deals with the data the user gave and transmits them to `extork`. The result of `extork`[12] is then included in a new predefined web page, which will display the table in a user-friendly way. There are four types of cells: the seed sentence (in red), attested sentences (in black), white cells and black cells (reflecting analogical impossibilities).



	☐	☐	☐	☐	☐	☐	☐
☐	I prefer Italian food.	Do you prefer Japanese food?	I prefer seafood.	I prefer French food.	I prefer Japanese food.	I'd prefer Japanese food.	I'd prefer Weste
☐	I like Italian food.	Do you like Japanese food?	I like seafood.		I like Japanese food.	I'd like Japanese food.	I'd like Western
☐	I'd like to have Italian food.		I'd like to have seafood.	I'd like to have French food.	I'd like to have Japanese food.		
☐	I like Italian food very much.				I like Japanese food very much		
☐	I'd like to eat some Italian food.		I'd like to eat some seafood.	I'd like to eat some French food	I'd like to eat some Japanese fc		
☐	I'd like to have some Italian foo			I'd like to have some French fo	I'd like to have some Japanese		
☐	How about Italian food?	■			How about Japanese food?		
☐	My favorite food is Italian.	■	■				
☐	Do you like Italian food?	■	Do you like seafood?	Do you like French food?	Do you like Japanese food?		
☐	How about some Italian food?	■			How about some Japanese foo		
☐	Is there a reasonable restauran	■		Is there a reasonable restauran	Is there a reasonable restauran		

Figure 5: The result page.

A few features of this page:

- Checkboxes on the top and left parts of the table allow the user to exchange two rows or two columns. This is helpful when trying to sort lines by hand in order to make certain types of organization or specific behaviours appear. But this is mainly a test feature.
- Sentences are data which are difficult to visualise in a compact way, and as the size of the cells could not be best fitted automatically [13], two buttons were introduced, one for reducing the size of the cells, and the other one for extending it. Presenting this data is not easy, because of the large amount of information it carries.
- When clicking in a cell containing a sentence, the `extork` program is called recursively in order to display the matrix associated with this sentence in a new window. This feature makes the exploration of the subcorpus possible.

(2.3.3) Servers and sockets

The next improvement concerned speed. With the method as it has been presented, `cat-btec` is executed each time a request is made, and many computations are performed again and again. Why should we call up the entire BTEC and use the `grep` command several times if we want to work on the same subcorpus?

The solution that was adopted relied on the use of servers and sockets. The `extork` program saw its behaviour modified: it does not read data from the standard input and

[12] In the form of a JavaScript file.

[13] Because font features cannot always be tuned by a program.

print its result to the standard output anymore. It is now initially launched as a server, which first stores the subcorpus data in memory, and then answers questions that any client asks it using sockets. Given a string, it sends back the associated table.

A library (developped during this internship and now included in the Aleph library[Lepage 92]) manages this task in order to simplify the complex use of sockets. Port number management is hidden, every server being characterized by a name. Three basic functions are available to the programmer:

- usrsktsend: sends a given message to a given server
- usrsktclose: used when a program need not talk to a given server anymore.
- usrsktwait: creates a server with a given name, which will execute a given function[14] on any message sent by a client.

We launch one server per subcorpus.

As a result, the behaviour of the CGI is as follows:

- get the data entered by the user
- search for a server corresponding to the subcorpus defined by the user
- if this server does not yet exist, launch it, and make it store information concerning the subcorpus defined by the user
- ask a question to the server
- get its answer and give it back to the web browser.

The gain is appreciable: using the HTML interface, while the matrix of *I like Japanese food.* took at least a minute to appear, it only took ten seconds if the server was already launched. When interrogated directly using little command-line clients, the server answers almost instantaneously ; the time it takes for the HTML table to appear is now caused almost entirely by JavaScript itself and cannot be reduced.

(2.3.4) Files necessary to make everything work

To use these programs, one will need:

- an operational web server. While the long-term objective is to make this program accessible through the Internet, right now the client and the server have to be on the same machine. The use of the HTML / JavaScript combination should make the migration easy. The one we used is Apache: it is free, efficient and easy to use. Such a server is necessary because of the use of CGIs, which are not managed by the browsers themselves.
- the `extork` program (which heavily uses the Aleph library), in the same directory as the CGI
- the CGI (named `matrice.cgi`), located in the CGI-BIN directory of the web server
- the `matrice.html` web page

[14] The signature of such a function is: char* fun(char*)

- the patron.html web page: as its name shows, this file is a pattern, filled by the CGI to display tables.
- the matrice.js file, which contains all of the JavaScript source code used in the two previous HTML documents.

For the purpose of testing, a few other tools are also available to talk directly to the servers.

3 Generation of new sentences

3.1 Principle

If it is for instance possible to have a corpus containing all the forms of regular verbs[15], and then making a table with no hole, this is not thinkable when dealing with sentences. Though the corpus we use is enormous, it does not contain all the possible utterances[16] related to the domain of travel. That is why the tables output by `extork` may sometimes be hollow. However, very often, these holes could be filled. In Fig. 3, one may feel that the third cell of the second column should be filled with *My favorite food is Japanese.*

Among its objectives, SLT focuses on making a large multilingual corpus available to the research community. This task can be done by hand (as has been the case until now), but we can see that the corpus itself contains enough information to automatically create new utterances. If we have a hundred sentences including the string *Japanese food*, should we again type a hundred sentences to talk about Chinese food?

In a certain sense, we are searching for elements in the convex envelope of the corpus.

This is a generalisation of analogy: if we put two commutation series that share a common phrase on orthogonal axes, the rectangle defined can be filled in automatically, exactly in the same way that children fill in multiplication tables.

Of course, the sentences produced may not always be correct. There are several levels of mistakes, corresponding to several levels of correctness (orthography, syntax, semantics, pragmatics, etc...) Languages are full of irregularities, or sub-rules: the past forms of *be*, *go* or *come* are not *beed*, *goed* or *comeed*, all forms that may be generated by analogy. These are considerations we do not take care of, because the only things we use are the analogy relation and the corpus itself. Our approach does not allow room for rules: we rely only on the organization and the alignments present in the corpus. Our approach relies on the idea that no extra information is required.

3.2 Implementation

In fact, the changes relative to what was described are very limited. Just before it gives its table, `extork` fills the empty cells thanks to an analogy equation solver present in the Aleph library. The computation time does not vary appreciably.

The problem here is that a perfect analogy solver has not yet been designed. This is another field of research not related here. An analogical equation may have several solutions; all of them must be displayed so that the user can choose the best ones, or none if they are all wrong.

Now `extork` would give back Fig. 6.

In this particular case, the $\mathcal{B}$ and $\mathcal{C}$ exhibit a relevant opposition, and the generated sentences are all correct. Not all results are that perfect.

3.3 Interface

By generating new sentences, the task becomes one of expanding the corpus, and consequently, a few features were added to the interface. The HTML form remains almost the

[15] The reason why this work deals essentially on the character level, and not the word level, is that we may want to be able to handle languages with a rich morphology, like Finnish. In such languages, an approach by dictionnary is impossible due to the almost infinite generative power of morphology

[16] The number of sentences in a language is infinite...

I prefer Italian food.	*I prefer Japanese food.*	*I prefer French food.*	*I prefer seafood.*
I like Italian food.	*I like Japanese food.*	**I like French food.**	*I like seafood.*
My favorite food is Italian.	**My favorite food is Japanese.**	**My favorite food is French.**	●
I'd like to have some Italian food.	*I'd like to have some Japanese food.*	*I'd like to have some French food.*	**I'd like to have some seafood.**

Figure 6: The table of Fig. 3 (associated with the sentence *I prefer Italian food.*), where generated sentences have been added (in bold font).

same. However, the page displaying the table underwent major changes. The main new functionality is that the user can indicate, among the proposed produced sentences, which ones are correct, and which ones are not; the result of these decisions is then saved on the server for further use.

Why should we rely on the user to determine if the generated utterances are correct or not? Simply because there is no other way. As we said, if a sentence is wrong, it can be at several levels. And even though low level ones could be detected automatically[17], there is no program today that is capable of saying wether sentences such as *I want to eat a cup of tea.*, *I like Italian food, and especially tempura.* or *Can I pay with eight-dollar bills?* should be added to the corpus or not. It only depends on what we want the corpus to contain: does it have to contain all grammatically correct sentence? Do they have to make sense? Do they have to be true? Only a human operator can solve these issues. And therefore an adapted interface is needed.

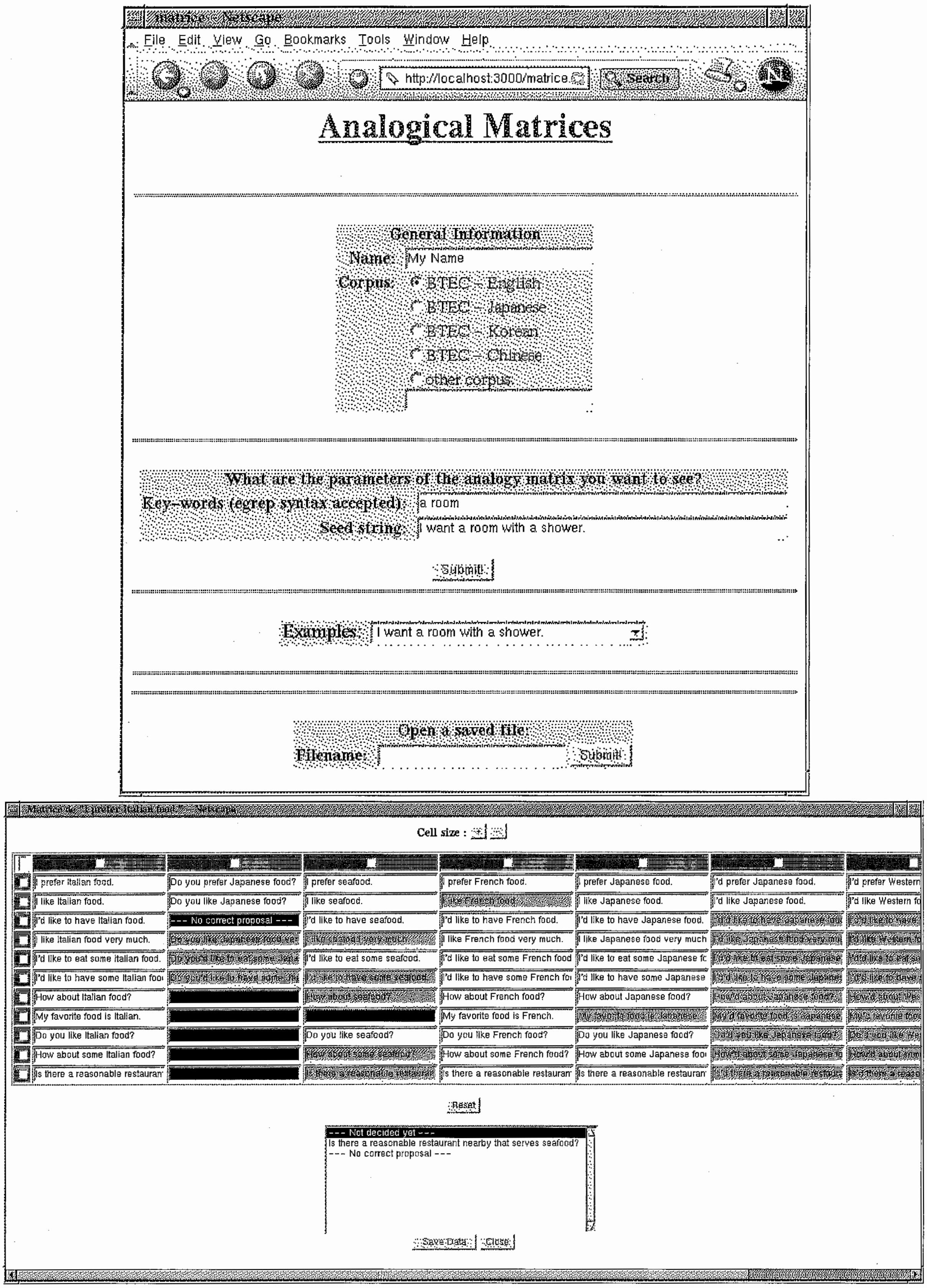
The changes undergone by the interface are shown in Fig. 7.

The added elements are:

- on the main page, the user may ask to visualize a previously saved matrix.
- on the table page, the cells that were previously empty now contain a suggested generated sentence. Their background color is turquoise.
- the box at the bottom of the page is updated each time the user clicks a cell. There, he may choose:
  - not decided yet: the default state, meaning no choice has been made.
  - sentences: the analogical equation solver may possibly propose several solutions in its further developments; if one of these sentences is correct and is to be added to the corpus, the user may select it by clicking. The text of the cell will be written in blue with a white background.
  - No correct solution: none of the suggestions is correct. The corresponding cell background color will turn black.
- At the end of the operation, everything is saved in a file on the server when the user clicks the "save data" button. A new CGI (`validation.cgi`) manages this task.

To summarize, the meaning of each color is as follows: White cells are correct sentences; black cells correspond to impossibilities; and turquoise are those the user has not done yet.

[17] Thanks to spell checkers or grammar checkers, even if they are not 100% accurate.


matrice - Netscape
File Edit View Go Bookmarks Tools Window Help
http://localhost:3000/matrice
Search
Submit!
Submit!
Matrice de "I prefer Italian food." - Netscape
Cell size :
Reset
Save Data
Close

Analogical Matrices

General Information

Name: My Name

Corpus:
● BTEC – English
○ BTEC – Japanese
○ BTEC – Korean
○ BTEC – Chinese
○ other corpus:

What are the parameters of the analogy matrix you want to see?

Key–words (egrep syntax accepted): a room

Seed string: I want a room with a shower.

Examples: I want a room with a shower.

Open a saved file:

Filename:

	□	□	□	□	□	□	□
□	I prefer Italian food.	Do you prefer Japanese food?	I prefer seafood.	I prefer French food.	I prefer Japanese food.	I'd prefer Japanese food.	I'd prefer Western
□	I like Italian food.	Do you like Japanese food?	I like seafood.	I like French food.	I like Japanese food.	I'd like Japanese food.	I'd like Western fo
□	I'd like to have Italian food.	--- No correct proposal ---	I'd like to have seafood.	I'd like to have French food.	I'd like to have Japanese food.	I'd'd like to have Japanese foo	I'd'd like to have
□	I like Italian food very much.	Do you like Japanese food ver	I like seafood very much.	I like French food very much.	I like Japanese food very much	I'd like Japanese food very mu	I'd like Western fo
□	I'd like to eat some Italian food.	Do you'd like to eat some Japa	I'd like to eat some seafood.	I'd like to eat some French food	I'd like to eat some Japanese fc	I'd'd like to eat some Japanese	I'd'd like to eat so
□	I'd like to have some Italian foo	Do you'd like to have some Ja	I'd like to have some seafood.	I'd like to have some French fo	I'd like to have some Japanese	I'd'd like to have some Japanes	I'd'd like to have
□	How about Italian food?		How about seafood?	How about French food?	How about Japanese food?	How'd about Japanese food?	How'd about We
□	My favorite food is Italian.			My favorite food is French.	My favorite food is Japanese.	My'd favorite food is Japanese	My'd favorite foo
□	Do you like Italian food?		Do you like seafood?	Do you like French food?	Do you like Japanese food?	Do'd you like Japanese food?	Do'd you like We
□	How about some Italian food?		How about some seafood?	How about some French food?	How about some Japanese foo	How'd about some Japanese fo	How'd about som
□	Is there a reasonable restauran		Is there a reasonable restauran	Is there a reasonable restauran	Is there a reasonable restauran	Is'd there a reasonable restau	Is'd there a reaso

--- Not decided yet ---
Is there a reasonable restaurant nearby that serves seafood?
--- No correct proposal ---

Figure 7: The previous general interface adapted to the generation of sentences.

3.4 Experiments and results

We could generate sentences from large symmetrical tables; but we believed that they would be more likely to be correct if they belonged to "good" paradigms. This is the assumption that we tried to verify in the following tests. The question is: do holes in the reduced paradigmatic table correspond more often to correct utterances than holes in the complete symmetric table?

To evaluate the efficiency of our method, we picked up 22 seed sentences at random from the BTEC[18]. As we said before, because this corpus is quite large, for each of our 22 seed sentences, we kept only several thousand sentences as a sub-corpus by filtering with a typical keyword of the chosen situation. The 22 seed sentences, their filter and the number of the sentences in the associated sub-corpora are listed in Table 1.

seed sentence label	filter (keyword)	size of the sub-corpus	seed sentence
20dollars	dollars	3035	*About Twenty dollars.*
CatchTaxi	taxi	1021	*Where can I catch a taxi?*
FeelBlue	feel	2296	*I feel blue.*
GetPostOffice	office	2333	*How do I get to the post office?*
HardTime	time	10485	*I had a hard time.*
JapFood	food	1660	*I like Japanese food.*
LeftTrain	train	2803	*I left something in the train.*
LetmeSee	Let me	1593	*Let me see it, please.*
OnBusiness	business	1639	*I'm on business.*
OutOffice	office	2333	*He is out of the office.*
PleaseTaxi	taxi	1021	*Please get me a taxi.*
PreferSeafood	food	1660	*I prefer seafood.*
SeePass	passport	573	*Can I see your passport, please.*
TakeTrain	train	2803	*Take the train.*
TrainTime	train	2803	*Will the train leave on time?*
WantRoom	a room	600	*I want to book a room.*
WantCoffee	coffee	1519	*I want a coffee.*
WhatPrice	price	1604	*What's the price?*
WhereOffice	office	2333	*Where is the office?*
WorkTrading	work	2061	*I work for a trading compagny.*
YenDollars	dollars	3035	*Please change yen into dollars.*
YouFeelTired	feel	2296	*Do you feel tired?*

Table 1: List of seed sentences used in the experiments.

We performed counting for the complete paradigm tables and for the new densified tables obtained automatically by a program. The results are shown in Table 2. On the left, the increase in the ratio of attested sentences over all theoretically possible sentences is to be interpreted as a measure of the increase of the paradigmatic density when densifying the paradigm table. In our experiments, the density has been more than doubled, from 7 % to 16 %. The possible loss caused by the densification is discussed later. On the right part of Table 2, the ratio between the number of new correct sentences and the total number of new generated sentences, which stands for the quality of the analogical generation, increases from 62 % to 70 %. Clearly, the densification of the paradigm tables using our method increases the reliability of the new generated sentences.

[18] Among those for which the associated table is not empty.

Going back to our initial goal, which is the automatic production of new sentences to be added to a corpus, during these reported experiments, we were able to generate 1,427 new correct sentences from only 22 seed sentences using complete paradigm tables. By densifying the paradigm tables, only $226/1,427 = 16$ % of the new correct sentences were left aside. In other words, the proportion of new correct sentences retained in the densified tables is 84 %. Combined with the increase in reliability, this confirms that densification prioritizess quality over quantity for the new generated sentences.

	complete table	densified table	complete table	densified table
seed sentence label	observed / possible sentences	observed / possible sentences	correct / all new sentences	correct / all new sentences
20dollars	62 / 1770 = 3 %	62 / 731 = 8 %	371 / 421 = 88 %	365 / 403 = 91 %
CatchTaxi	16 / 171 = 9 %	16 / 70 = 22 %	23 / 28 = 82 %	18 / 21 = 86 %
FeelBlue	40 / 630 = 6 %	33 / 224 = 14 %	51 / 86 = 59 %	40 / 63 = 64 %
GetPostOffice	19 / 136 = 13 %	19 / 42 = 45 %	14 / 21 = 67 %	14 / 21 = 67 %
HardTime	8 / 66 = 12 %	6 / 35 = 17 %	22 / 29 = 76 %	13 / 15 = 87 %
JapFood	50 / 820 = 6 %	43 / 418 = 10 %	178 / 468 = 38 %	137 / 290 = 47 %
LeftTrain	15 / 105 = 14 %	14 / 36 = 38 %	13 / 16 = 81 %	11 / 14 = 79 %
LetmeSee	32 / 595 = 5 %	22 / 304 = 7 %	26 / 50 = 52 %	19 / 22 = 86 %
OnBusiness	12 / 66 = 18 %	10 / 36 = 27 %	14 / 39 = 36 %	9 / 26 = 35 %
OutOffice	30 / 435 = 6 %	27 / 81 = 33 %	26 / 83 = 31 %	19 / 46 = 41 %
PleaseTaxi	57 / 946 = 6 %	53 / 475 = 11 %	111 / 157 = 71 %	83 / 120 = 69 %
PreferSeafood	27 / 153 = 17 %	27 / 77 = 35 %	26 / 47 = 55 %	26 / 42 = 62 %
SeePass	20 / 171 = 11 %	19 / 78 = 24 %	40 / 49 = 82 %	36 / 41 = 88 %
TakeTrain	20 / 231 = 8 %	20 / 57 = 35 %	39 / 39 = 100 %	37 / 37 = 100 %
TrainTime	21 / 276 = 7 %	19 / 44 = 43 %	11 / 25 = 44 %	10 / 24 = 42 %
WantRoom	8 / 45 = 17 %	8 / 25 = 32 %	6 / 6 = 100 %	6 / 6 = 100 %
WantCoffee	39 / 351 = 11 %	37 / 126 = 29 %	30 / 53 = 57 %	25 / 46 = 54 %
WhatPrice	26 / 378 = 6 %	24 / 195 = 12 %	157 / 272 = 58 %	107 / 170 = 63 %
WhereOffice	18 / 231 = 7 %	14 / 121 = 11 %	77 / 136 = 57 %	52 / 82 = 64 %
WorkTrading	27 / 435 = 6 %	25 / 104 = 24 %	85 / 93 = 91 %	69 / 73 = 95 %
YenDollars	30 / 351 = 8 %	29 / 182 = 15 %	64 / 149 = 43 %	62 / 104 = 60 %
YouFeelTired	30 / 253 = 11 %	30 / 76 = 39 %	43 / 48 = 90 %	43 / 46 = 94 %
average	607 / 8,615 = 7 %	557 / 3,537 = 16 %	1,427 / 2,315 = 62 %	1,201 / 1,712 = 70 %

Table 2: Results for 22 seed sentences.

4 Application to translation

4.1 Principle

In [Lepage 03], Lepage describes a method of translation using analogy. This is what we tried to implement here. We have a large aligned bicorpus: the method used is example-based, and may work quite well with sentences similar to sentences of the bicorpus. The global idea is explained in Fig. 8.

日本料理が好きです。	:	メキシコ料理が好きです。	=	日本料理がいいんですが。	: x	⇒	x =	メキシコ料理がいいんですが。
↑		↑		↑				↑
I like Japanese food.	:	*I like Mexican food.*	=	*I prefer Japanese food.*	:			**I prefer Mexican food.**

Figure 8: Principle of translation by analogy

Fig. 8, which represents a translation “cube”, shows how suggestions are made. We first search for all analogies involving the sentence to be translated, here, *I prefer Mexican food.* Let us consider one of these analogies. Then for the three other sentences of this analogy, which belong to the corpus, the corresponding translations are taken (this operation is represented by arrows); the solution of the analogical equation that is formed in the second domain is our suggestion (here, メキシコ料理がいいんですが, which is correct).

This idea is extended to a whole paradigm. The succesive objects of the process are now:

- the sentence to be translated
- the paradigm of this sentence (computed by `extork`)
- the translation term to term of this table (which forms a similarly shaped table). This is not necessarily a paradigm.
- a series of weighted suggested translations for the user input.

One point we must consider is that translation is not a bijection. This is so, first because of paraphrasing, and secondly because a crucial distinction in one language may not necessarily be expressed in another one, or in a different manner. The Japanese “はい” is translated into English by “yes”, or “no”, depending on the affirmative or negative polarization of the question sentence[19].

So the simple cube shown earlier hides a more complex issue. For the translation of a single utterance, the number of such cubes is multiplied because:

- there may be many (B,C,D) triples in the first domain such as $A : B = C : D$
- each B, C or D may have several translations.

[19] conversely, “yes” is translated by either “はい” or “いいえ”.

Moreover, we would naturally like to take into consideration the frequency information present in the corpus: just knowing that "yes" may be translated into "はい" or "いいえ" gives fewer clues than knowing that in our corpus, this first translation appears 100 times and the second one only 10 times.

Consequently, many parameters should be taken into account. The issue of how to deal with their relative importance is still open.

4.2 Implementation

To perform translation, we use a method similar to object oriented programming; independent modules are launched, and they communicate through sockets.

These modules are:

- Domain servers. They contain all the utterances of a corpus: the data they contain is the list of these utterances.
- Correspondence servers. There is one per pair of languages; they know how any sentence of the first corpus translates in the second and vice versa. As said before, because translation is not a bijection, a sentence in a corpus may be translated in several ways in another language. Consequently, the data that such servers contain is the list of pairs of sentences, each pair weighted by the number of time this link is present in the bicorpus.

Suppose we wanted to translate from English into Japanese. Before any request can be made, a domain server must be launched and filled for each corpus; also a correspondence server must be launched with the bicorpus as its input parameter.

Then a PROGRAM using these modules can be launched. Things will happen this way: the user first inputs a sentence he wants to be translated, as well as a filter. The PROGRAM transmits both these pieces of information to the English language domain server, which will give back as a result the subcorpus corresponding to the filter given. Then the PROGRAM will launch `extork` on this subcorpus; it will then obtain a table[20]. Next, thanks to the correspondence server, this table will be translated into Japanese, except, of course, for the A cell, which does not correspond to anything in the second domain. Ideally, the translation of A would be the unique sentence which would be the solution of the corresponding $x : B_i = C_j : D_{ij}$ equations; but as there may be many such equations, a criterion must be found (possibly based on frequencies).

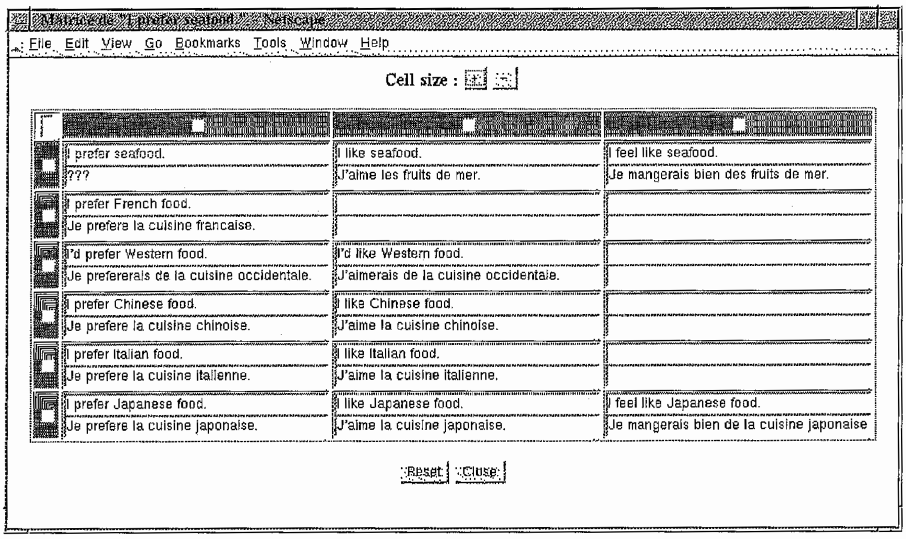
4.3 Interface

From the corpus expansion task to the translation task, the main page does not change; however, the result page does not represent the same data as in Section 3.3.

Here our formal objective is not to present a table but a translation; nevertheless, the result page presents an analogical table whose goal is to show how the suggested translation was deduced.

The two tables (the paradigm associated to the sentence to be translated, and the table translated from the first one) are presented together: each cell contains both an English sentence and its French translation just below.

[20] the table obtained does only contain attested sentences, no generated ones.


Matrice de "I prefer seafood." - Netscape
File Edit View Go Bookmarks Tools Window Help
Cell size :
Reset
Close

I prefer seafood. ???	I like seafood. J'aime les fruits de mer.	I feel like seafood. Je mangerais bien des fruits de mer.
I prefer French food. Je prefere la cuisine francaise.		
I'd prefer Western food. Je prefererais de la cuisine occidentale.	I'd like Western food. J'aimerais de la cuisine occidentale.	
I prefer Chinese food. Je prefere la cuisine chinoise.	I like Chinese food. J'aime la cuisine chinoise.	
I prefer Italian food. Je prefere la cuisine italienne.	I like Italian food. J'aime la cuisine italienne.	
I prefer Japanese food. Je prefere la cuisine japonaise.	I like Japanese food. J'aime la cuisine japonaise.	I feel like Japanese food. Je mangerais bien de la cuisine japonaise

Figure 9: The translation presentation page. The suggested translation should be located where the *???* are.

The table of Fig. 9 was obtained with data from the English BTEC and French data obtained using a commercial machine translation system in an experiment performed during July 2003.

4.4 Results

As the link between the translation system and the interface is not completed yet, the following results were obtained with command-line programs. Here are what the client and the server sent back when asked for a translation (here, three simple English sentences we want in Japanese):

1. *I prefer Mexican food.*
   - 2 メキシコ料理のほうが好きです。 *(correct)*
   - 2 メキシコ料理がいいんですが。 *(correct)*
   - 1 メキシコ料理を食べてみい燭い列ですが。 *(incorrect)*
2. *Shall I wait?*
   - 1 待ちますか。 *(correct)*
3. *Can I have your ticket, please?*
   - 1 切符を拝見できますか。 *(correct)*
   - 1 切符を拝見します。 *(correct)*
   - 1 切符を見せてください。 *(correct)*,

where the number before each suggestion is the frequency of the corresponding result, *i.e.*, the number of translation squares that generated it.

The traces of the server indicate what analogies were found, and with which analogies the results were produced. These are in Annex ??.

5 Conclusion

The main goal of this work was to show that simple and linguistically justified concepts may give interesting results when applied to some Natural Language Processing tasks.

We showed that without making any hypothesis on the domain we work on, it was possible to automatically extract structures; and these structures permit us to expand a corpus, or to translate utterances from one language to another language.

Still, it confirms the idea that processes cannot be totally automated, because of the complexity of natural languages.

A few ideas that should be inspected during future research are as follows:

- In the paradigms we extract, the first row and column are always full, by definition of the construction method. However, there is no reason for that; the possibility of having blank cells surely may improve the relevance of the tables obtained.
- The current communication system, based on sockets, should probably be replaced by a DBMS system. Many problems were faced when trying to implement the use of servers to make our programs communicate.
- As it was mentioned earlier, the analogy definition used here is between strings of symbols. This is a path that could be studied too: is there a better level from which to look for analogy? The advantage of the choice made in our research is that it does not make any assumption on the language; we just have to know into how many bytes a character is encoded.

付録 A　A few generation tables

Here are a few examples of what may be generated using our interface.

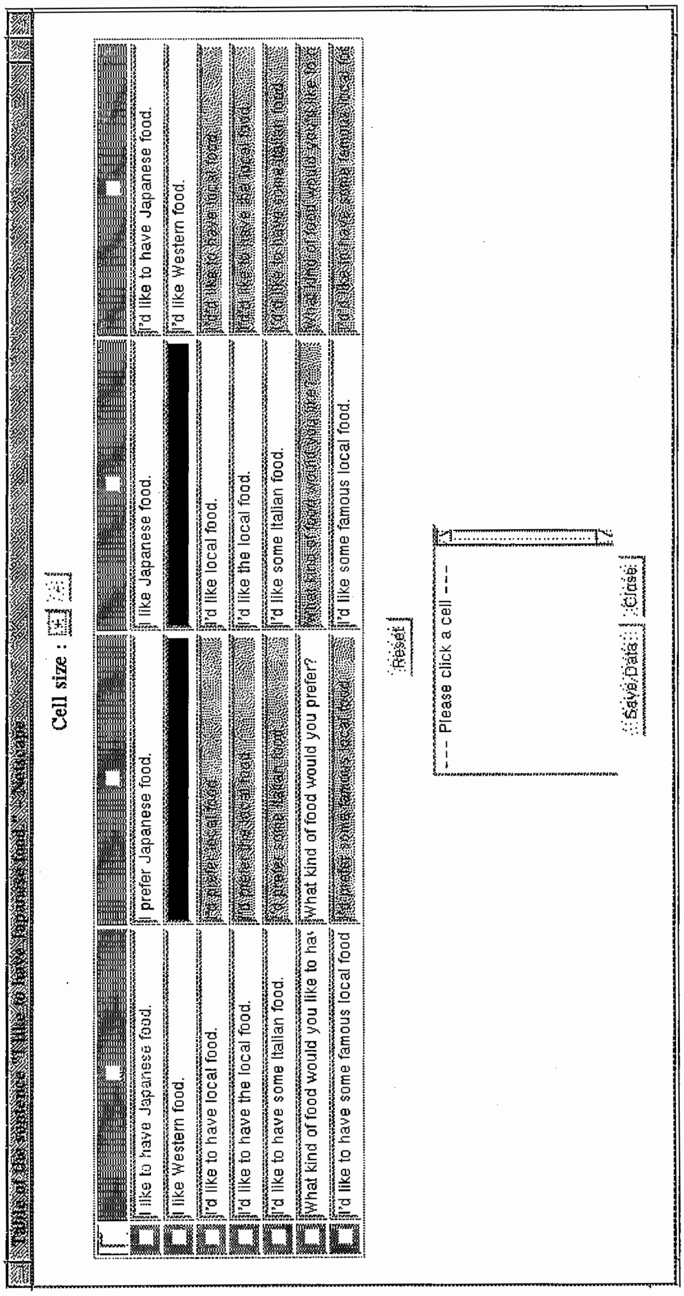


Table of the sentence "I like to have Japanese food." - Netscape

Cell size :

□	I like to have Japanese food.	I prefer Japanese food.	I like Japanese food.	I'd like to have Japanese food.
□	I like Western food.			I'd like Western food.
□	I'd like to have local food.	I'd prefer local food.	I'd like local food.	I'd'd like to have local food.
□	I'd like to have the local food.	I'd prefer the local food.	I'd like the local food.	I'd'd like to have the local food.
□	I'd like to have some Italian food.	I'd prefer some Italian food.	I'd like some Italian food.	I'd'd like to have some Italian food.
□	What kind of food would you like to hav	What kind of food would you prefer?	What kind of food would you like?	What kind of food would you'd like to c
□	I'd like to have some famous local food	I'd prefer some famous local food.	I'd like some famous local food.	I'd'd like to have some famous local foc

Figure 10: I like to have Japanese food.

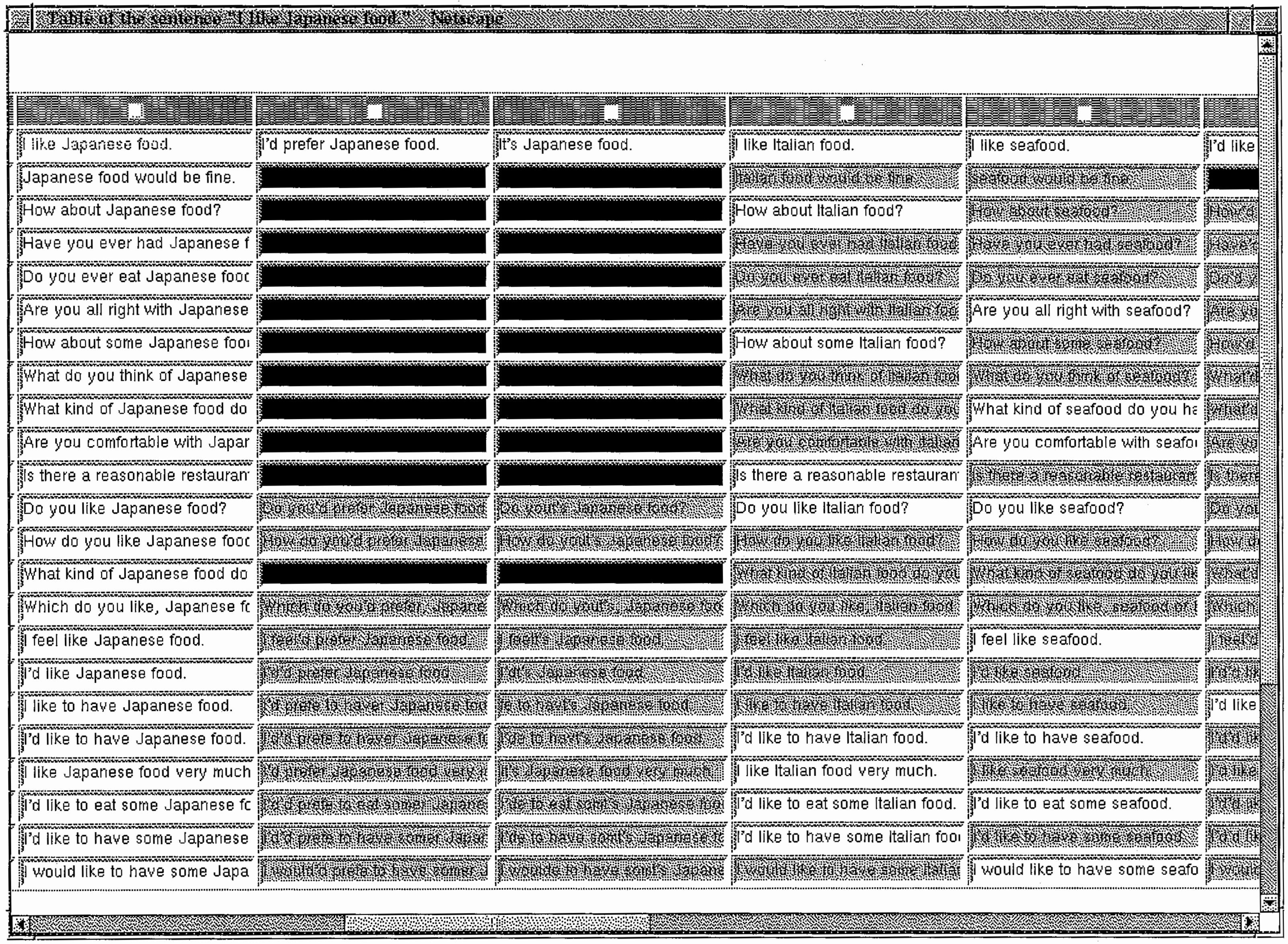
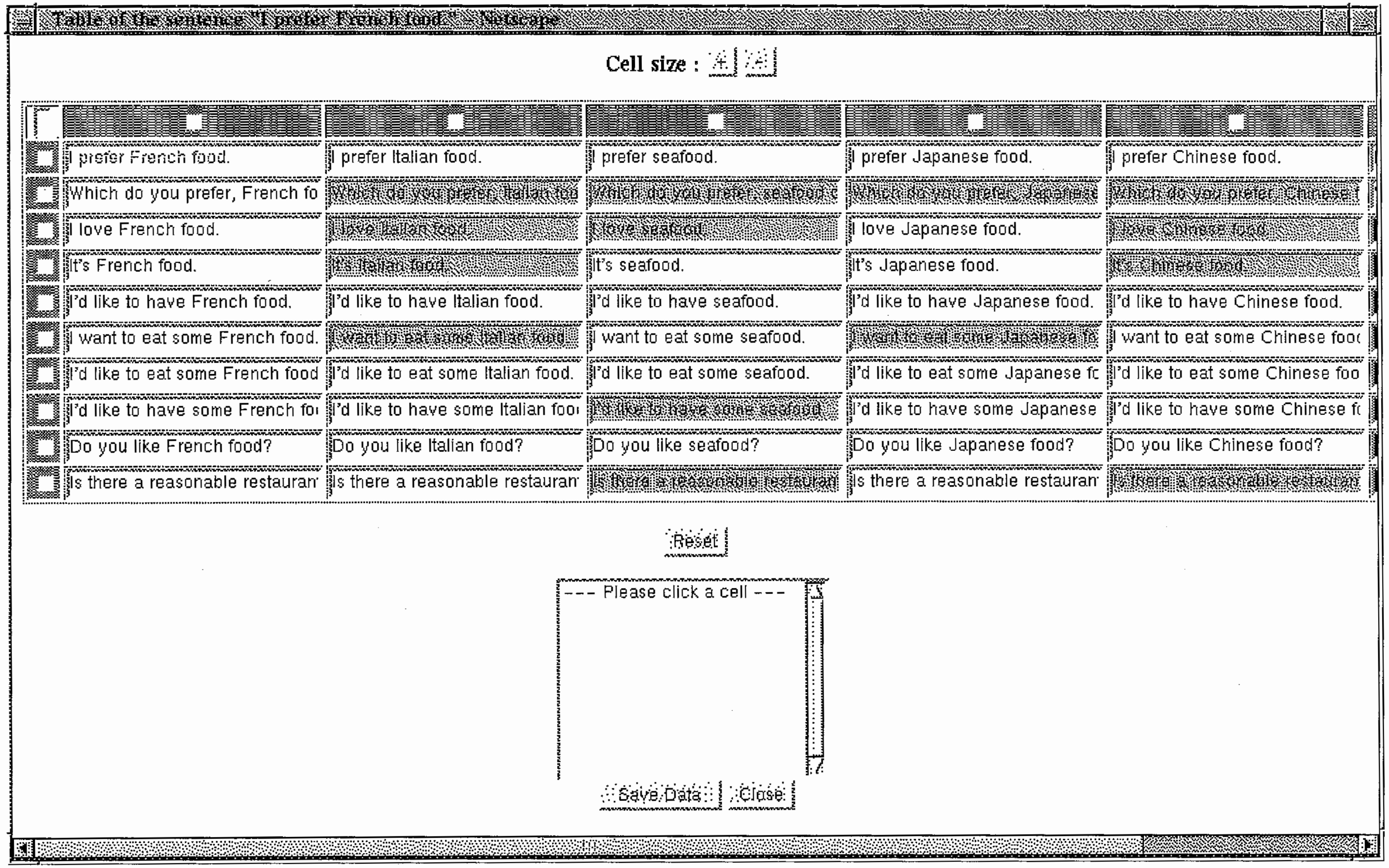


Table of the sentence "I like Japanese food." - Netscape

□	□	□	□	□	
I like Japanese food.	I'd prefer Japanese food.	It's Japanese food.	I like Italian food.	I like seafood.	I'd like
Japanese food would be fine.	[redacted]	[redacted]	Italian food would be fine.	seafood would be fine.	[redacted]
How about Japanese food?	[redacted]	[redacted]	How about Italian food?	How about seafood?	How'd
Have you ever had Japanese f	[redacted]	[redacted]	Have you ever had Italian food	Have you ever had seafood?	Have'
Do you ever eat Japanese fooc	[redacted]	[redacted]	Do you ever eat Italian food?	Do you ever eat seafood?	Do'd y
Are you all right with Japanese	[redacted]	[redacted]	Are you all right with Italian foo	Are you all right with seafood?	Are yo
How about some Japanese foo	[redacted]	[redacted]	How about some Italian food?	How about some seafood?	How'd
What do you think of Japanese	[redacted]	[redacted]	What do you think of Italian foo	What do you think of seafood?	What'd
What kind of Japanese food do	[redacted]	[redacted]	What kind of Italian food do you	What kind of seafood do you ha	What'd
Are you comfortable with Japar	[redacted]	[redacted]	Are you comfortable with Italian	Are you comfortable with seafo	Are yo
Is there a reasonable restauran	[redacted]	[redacted]	Is there a reasonable restauran	Is there a reasonable restauran	Is there
Do you like Japanese food?	Do you'd prefer Japanese food	Do you't's Japanese food?	Do you like Italian food?	Do you like seafood?	Do you
How do you like Japanese fooc	How do you'd prefer Japanese	How do you't's Japanese food?	How do you like Italian food?	How do you like seafood?	How d
What kind of Japanese food do	[redacted]	[redacted]	What kind of Italian food do you	What kind of seafood do you lik	What'd
Which do you like, Japanese fc	Which do you'd prefer, Japane	Which do you't's, Japanese foo	Which do you like, Italian food	Which do you like, seafood or I	Which
I feel like Japanese food.	I feel'd prefer Japanese food	I feel't's Japanese food	I feel like Italian food	I feel like seafood.	I feel'd
I'd like Japanese food.	I'd'd prefer Japanese food	I'd't's Japanese food	I'd like Italian food	I'd like seafood	I'd'd lik
I like to have Japanese food.	I'd prefe to haver Japanese foo	fe to have't's Japanese food	I like to have Italian food	I like to have seafood	I'd like
I'd like to have Japanese food.	I'd'd prefe to haver Japanese fc	I'de to have't's Japanese food	I'd like to have Italian food.	I'd like to have seafood.	I'd'd lik
I like Japanese food very much	I'd prefer Japanese food very n	It's Japanese food very much	I like Italian food very much.	I like seafood very much	I'd like
I'd like to eat some Japanese fc	I'd'd prefe to eat somer Japane	I'de to eat somt's Japanese foo	I'd like to eat some Italian food.	I'd like to eat some seafood.	I'd'd lik
I'd like to have some Japanese	I'd'd prefe to have somer Japar	I'de to have somt's Japanese fc	I'd like to have some Italian foo	I'd like to have some seafood	I'd'd lik
I would like to have some Japa	I would'd prefe to have somer J	I woulde to have somt's Japane	I would like to have some Italiar	I would like to have some seafo	I would

Figure 11: I like Japanese food.



	□	□	□	□	□
□	I prefer French food.	I prefer Italian food.	I prefer seafood.	I prefer Japanese food.	I prefer Chinese food.
□	Which do you prefer, French fo	Which do you prefer, Italian foo	Which do you prefer, seafood c	Which do you prefer, Japanese	Which do you prefer, Chinese f
□	I love French food.	I love Italian food.	I love seafood.	I love Japanese food.	I love Chinese food.
□	It's French food.	It's Italian food.	It's seafood.	It's Japanese food.	It's Chinese food.
□	I'd like to have French food.	I'd like to have Italian food.	I'd like to have seafood.	I'd like to have Japanese food.	I'd like to have Chinese food.
□	I want to eat some French food.	I want to eat some Italian food.	I want to eat some seafood.	I want to eat some Japanese fo	I want to eat some Chinese food
□	I'd like to eat some French food	I'd like to eat some Italian food.	I'd like to eat some seafood.	I'd like to eat some Japanese fc	I'd like to eat some Chinese foo
□	I'd like to have some French fo	I'd like to have some Italian foo	I'd like to have some seafood.	I'd like to have some Japanese	I'd like to have some Chinese fc
□	Do you like French food?	Do you like Italian food?	Do you like seafood?	Do you like Japanese food?	Do you like Chinese food?
□	Is there a reasonable restauran	Is there a reasonable restauran	Is there a reasonable restauran	Is there a reasonable restauran	Is there a reasonable restauran

Figure 12: I prefer French food.

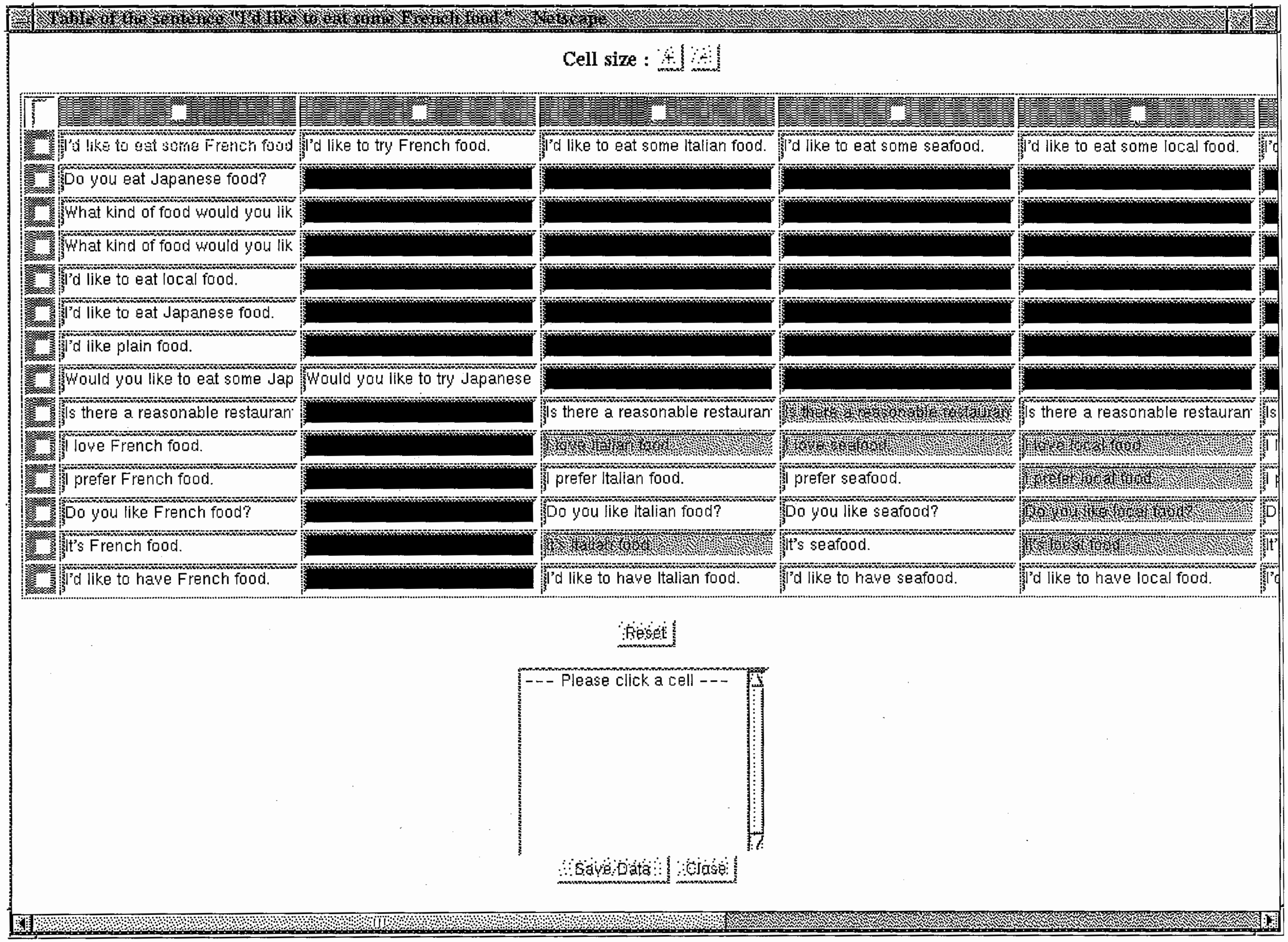


Table of the sentence "I'd like to eat some French food." - Netscape
Cell size : + -
Reset
Save Data
Close

--- Please click a cell ---

□	□	□	□	□	□	
□	I'd like to eat some French food	I'd like to try French food.	I'd like to eat some Italian food.	I'd like to eat some seafood.	I'd like to eat some local food.	I'c
□	Do you eat Japanese food?	■	■	■	■	■
□	What kind of food would you lik	■	■	■	■	■
□	What kind of food would you lik	■	■	■	■	■
□	I'd like to eat local food.	■	■	■	■	■
□	I'd like to eat Japanese food.	■	■	■	■	■
□	I'd like plain food.	■	■	■	■	■
□	Would you like to eat some Jap	Would you like to try Japanese	■	■	■	■
□	Is there a reasonable restauran	■	Is there a reasonable restauran	Is there a reasonable restauran	Is there a reasonable restauran	Is
□	I love French food.	■	I love Italian food.	I love seafood.	I love local food.	I l
□	I prefer French food.	■	I prefer Italian food.	I prefer seafood.	I prefer local food.	I p
□	Do you like French food?	■	Do you like Italian food?	Do you like seafood?	Do you like local food?	D
□	It's French food.	■	It's Italian food.	It's seafood.	It's local food.	It'
□	I'd like to have French food.	■	I'd like to have Italian food.	I'd like to have seafood.	I'd like to have local food.	I'c

Figure 13: I'd like to eat some French food.

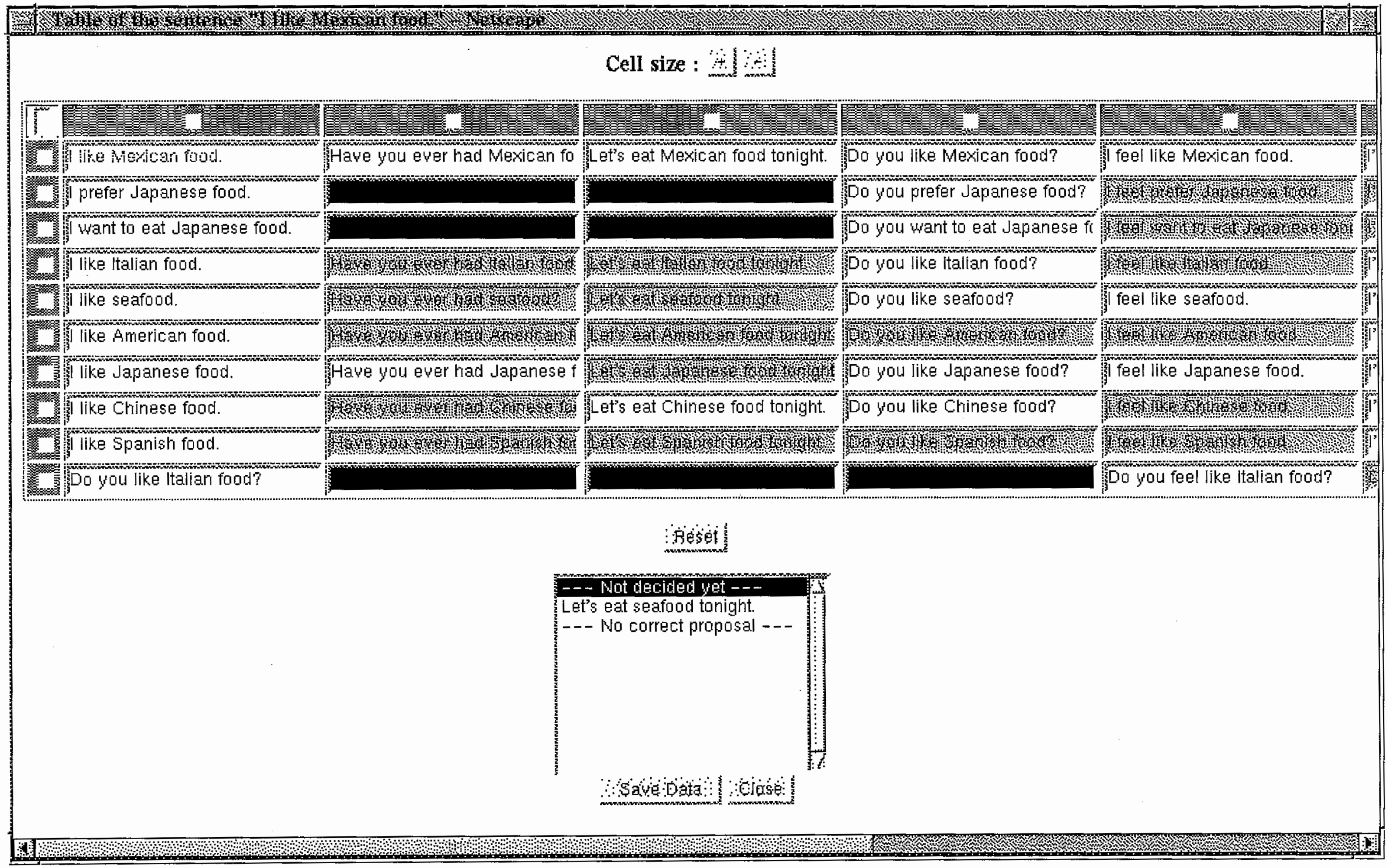


Table of the sentence "I like Mexican food." - Netscape

Cell size : + -

	□	□	□	□	□	
□	I like Mexican food.	Have you ever had Mexican fo	Let's eat Mexican food tonight.	Do you like Mexican food?	I feel like Mexican food.	I'
□	I prefer Japanese food.			Do you prefer Japanese food?	I feel prefer Japanese food.	I'
□	I want to eat Japanese food.			Do you want to eat Japanese fc	I feel want to eat Japanese foo	I'
□	I like Italian food.	Have you ever had Italian food	Let's eat Italian food tonight.	Do you like Italian food?	I feel like Italian food.	I'
□	I like seafood.	Have you ever had seafood?	Let's eat seafood tonight.	Do you like seafood?	I feel like seafood.	I'
□	I like American food.	Have you ever had American f	Let's eat American food tonight.	Do you like American food?	I feel like American food.	I'
□	I like Japanese food.	Have you ever had Japanese f	Let's eat Japanese food tonight	Do you like Japanese food?	I feel like Japanese food.	I'
□	I like Chinese food.	Have you ever had Chinese fo	Let's eat Chinese food tonight.	Do you like Chinese food?	I feel like Chinese food.	I'
□	I like Spanish food.	Have you ever had Spanish fo	Let's eat Spanish food tonight.	Do you like Spanish food?	I feel like Spanish food.	I'
□	Do you like Italian food?				Do you feel like Italian food?	

Reset

--- Not decided yet ---
Let's eat seafood tonight.
--- No correct proposal ---

Save Data | Close

Figure 14: I like Mexican food.

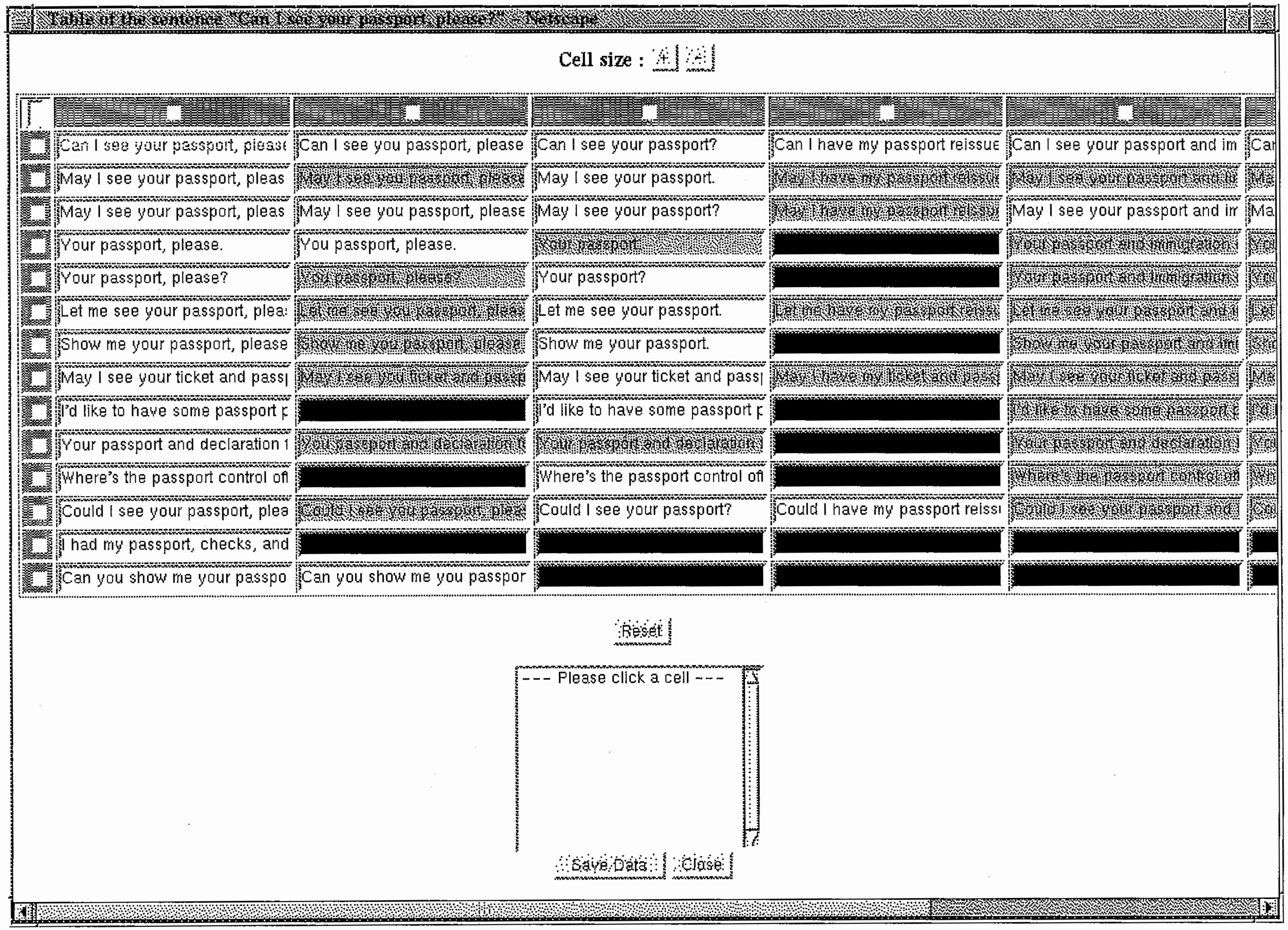


Table of the sentence "Can I see your passport, please?" - Netscape
Cell size :

	□	□	□	□	□	□
□	Can I see your passport, please	Can I see you passport, please	Can I see your passport?	Can I have my passport reissue	Can I see your passport and im	Car
□	May I see your passport, pleas	May I see you passport, please	May I see your passport.	May I have my passport reissu	May I see your passport and im	Ma
□	May I see your passport, pleas	May I see you passport, please	May I see your passport?	May I have my passport reissu	May I see your passport and im	Ma
□	Your passport, please.	You passport, please.	Your passport.	■	Your passport and immigration	Yo
□	Your passport, please?	You passport, please?	Your passport?	■	Your passport and immigration	Yo
□	Let me see your passport, plea:	Let me see you passport, pleas	Let me see your passport.	Let me have my passport reiss	Let me see your passport and i	Let
□	Show me your passport, please	Show me you passport, please	Show me your passport.	■	Show me your passport and imi	Sho
□	May I see your ticket and pass	May I see you ticket and passp	May I see your ticket and pass	May I have my ticket and pass	May I see your ticket and pass	Ma
□	I'd like to have some passport p	■	I'd like to have some passport p	■	I'd like to have some passport p	I'd
□	Your passport and declaration t	You passport and declaration t	Your passport and declaration	■	Your passport and declaration	Yo
□	Where's the passport control ofi	■	Where's the passport control ofi	■	Where's the passport control of	Wh
□	Could I see your passport, plea	Could I see you passport, plea	Could I see your passport?	Could I have my passport reiss	Could I see your passport and	Co
□	I had my passport, checks, and	■	■	■	■	■
□	Can you show me your passpo	Can you show me you passpor	■	■	■	■

Figure 15: Can I see your passport, please.

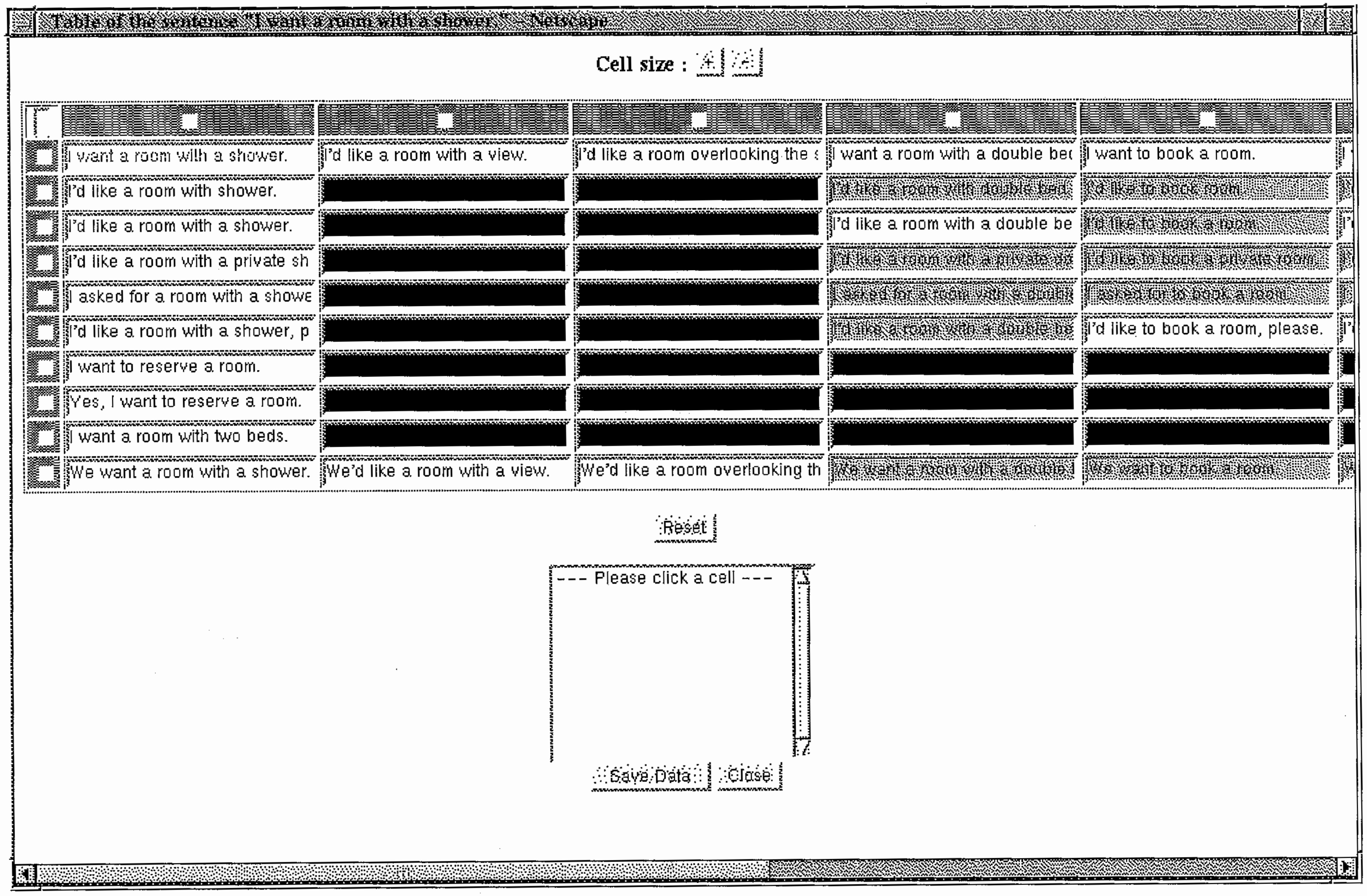


Table of the sentence "I want a room with a shower." - Netscape

Cell size : + -

□	□	□	□	□	□	
□	I want a room with a shower.	I'd like a room with a view.	I'd like a room overlooking the s	I want a room with a double bec	I want to book a room.	I
□	I'd like a room with shower.	■	■	I'd like a room with double bed.	I'd like to book room.	I
□	I'd like a room with a shower.	■	■	I'd like a room with a double be	I'd like to book a room.	I'
□	I'd like a room with a private sh	■	■	I'd like a room with a private do	I'd like to book a private room.	I
□	I asked for a room with a showe	■	■	I asked for a room with a doubl	I asked for to book a room.	I
□	I'd like a room with a shower, p	■	■	I'd like a room with a double be	I'd like to book a room, please.	I'
□	I want to reserve a room.	■	■	■	■	■
□	Yes, I want to reserve a room.	■	■	■	■	■
□	I want a room with two beds.	■	■	■	■	■
□	We want a room with a shower.	We'd like a room with a view.	We'd like a room overlooking th	We want a room with a double l	We want to book a room.	W

Reset

--- Please click a cell ---

Save Data | Close

Figure 16: I want a room with a shower.

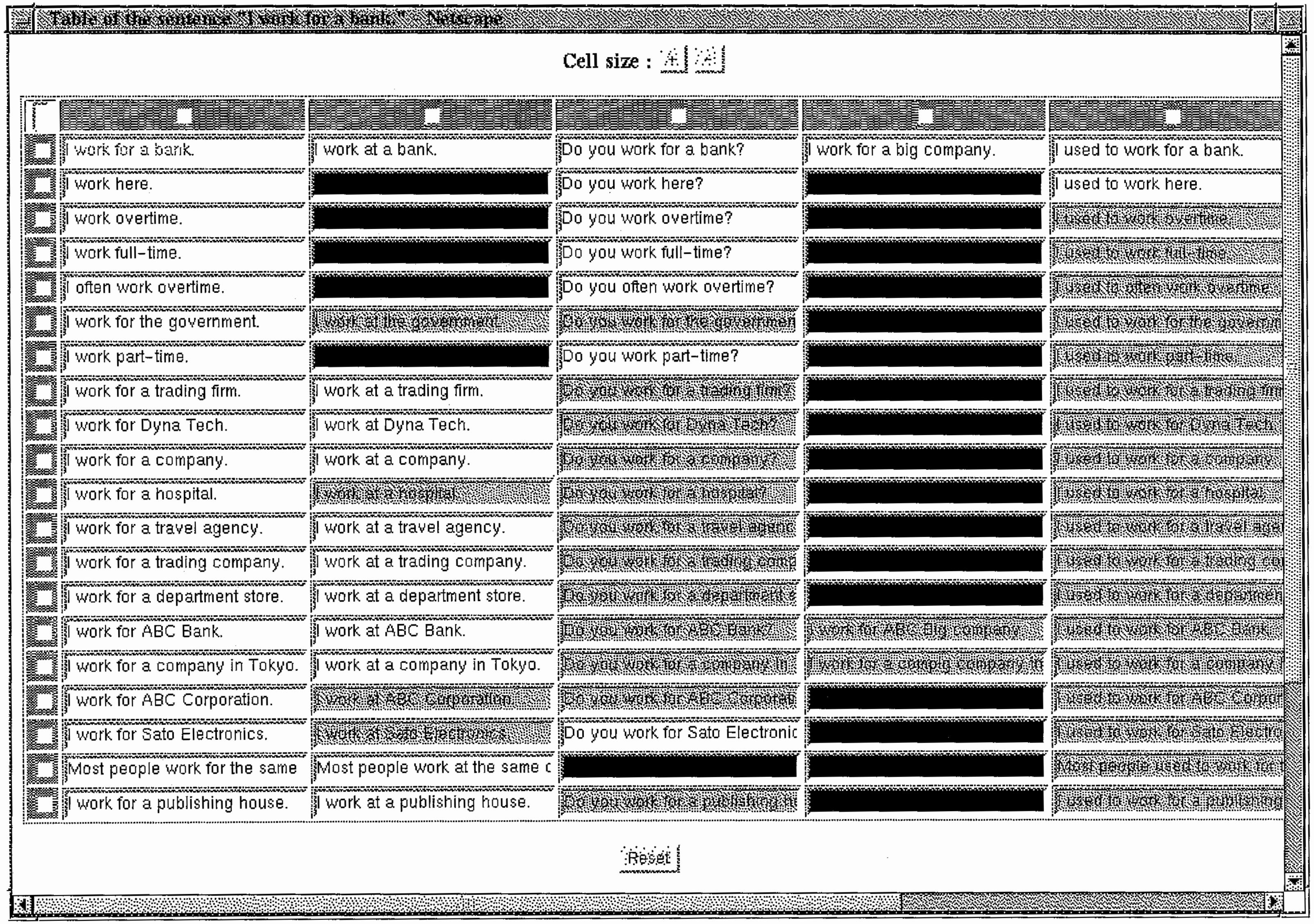


Table of the sentence "I work for a bank." - Netscape

Cell size :

	□	□	□	□	□
□	I work for a bank.	I work at a bank.	Do you work for a bank?	I work for a big company.	I used to work for a bank.
□	I work here.		Do you work here?		I used to work here.
□	I work overtime.		Do you work overtime?		I used to work overtime.
□	I work full-time.		Do you work full-time?		I used to work full-time.
□	I often work overtime.		Do you often work overtime?		I used to often work overtime.
□	I work for the government.	I work at the government.	Do you work for the governmen		I used to work for the governm
□	I work part-time.		Do you work part-time?		I used to work part-time.
□	I work for a trading firm.	I work at a trading firm.	Do you work for a trading firm?		I used to work for a trading firm
□	I work for Dyna Tech.	I work at Dyna Tech.	Do you work for Dyna Tech?		I used to work for Dyna Tech.
□	I work for a company.	I work at a company.	Do you work for a company?		I used to work for a company.
□	I work for a hospital.	I work at a hospital.	Do you work for a hospital?		I used to work for a hospital.
□	I work for a travel agency.	I work at a travel agency.	Do you work for a travel agenc		I used to work for a travel age
□	I work for a trading company.	I work at a trading company.	Do you work for a trading comp		I used to work for a trading co
□	I work for a department store.	I work at a department store.	Do you work for a department s		I used to work for a departmen
□	I work for ABC Bank.	I work at ABC Bank.	Do you work for ABC Bank?	I work for ABC Big company.	I used to work for ABC Bank.
□	I work for a company in Tokyo.	I work at a company in Tokyo.	Do you work for a company in	I work for a compig company in	I used to work for a company i
□	I work for ABC Corporation.	I work at ABC Corporation.	Do you work for ABC Corporati		I used to work for ABC Corpor
□	I work for Sato Electronics.	I work at Sato Electronics.	Do you work for Sato Electronic		I used to work for Sato Electro
□	Most people work for the same	Most people work at the same c			Most people used to work for t
□	I work for a publishing house.	I work at a publishing house.	Do you work for a publishing h		I used to work for a publishing

Reset

Figure 17: I work for a bank.

付録 B Some translation results

This technique seems promising, though it must be tested extensively. Below, the analogies in the first server are written, as well as the corresponding analogical four-tuple in Japanese. “” represents analogical impossibilities: the corresponding cube did not permit us to make a suggestion.

1. I prefer Mexican food:

   - *I prefer Mexican food. : I prefer Italian food. = I'd like to try some Mexican food. : I'd like to try some Italian food.*

     メキシコ料理を食べてみい煬い列ですが。 : イタリア料理がいいんですが。 = メキシコ料理を食べてみたいのです。 : イタリア料理がいいです

   - *I prefer Mexican food. : I prefer Italian food. = Is there a Mexican restaurant around here? : Is there a Italian restaurant arc*

     メキシコ料理がいいんですが。 : イタリア料理がいいんですが。 = この辺りにメキシコ料理店はありますか。 : この辺りにイタリア料理店はあ

   - *I prefer Mexican food. : I prefer Japanese food. = I like Mexican food. : I like Japanese food.*
     No solution : 日本料理がいいんですが。 = メキシコ料理が好きです。 : 日本食が好きです。

   - *I prefer Mexican food. : I prefer Japanese food. = Is there a Mexican restaurant around here? : Is there a Japanese restauran*

     メキシコ料理がいいんですが。 : 日本料理がいいんですが。 = この辺りにメキシコ料理店はありますか。 : この辺りに日本料理店はありますか

   - *I prefer Mexican food. : I like Mexican food. = I prefer Chinese food. : I like Chinese food.*
     メキシコ料理のほうが好きです。 : メキシコ料理が好きです。 = 中華料理のほうが好きです。 : 中華料理が好きです。

   - *I prefer Mexican food. : I prefer Chinese food. = Is there a Mexican restaurant around here? : Is there a Chinese restaurant ,*

     メキシコ料理のほうが好きです。 : 中華料理のほうが好きです。 = この辺りにメキシコ料理店はありますか。 : この辺りに中華料理店はありま

2. Shall I wait?

   - *Shall I wait? : I'll wait. = Shall I call an ambulance? : I'll call an ambulance.*
     待ちますか。 : 待ちます。 = 救急車を呼びましょうか。 : 救急車を呼びましょう。
   - *Shall I wait? : I'll wait. = Shall I ask him to call you back? : I'll ask him to call you back.*
     No solution : 待ちます。 = お電話するように申しましょうか。 : 折り返しこちらから電話させます。
   - *Shall I wait? : Please wait. = Shall I wrap it? : Please wrap it.*
     No solution : ちょっと待ってください。 = お包みしましょうか。 : 包んでください。
   - *Shall I wait? : Please wait a moment. = Shall I hold the line? : Please hold the line a moment.*
     No solution : しばらくお待ちください。 = このまま電話を切らずに待つのですか。 : しばらくお待ちくださいませ。
   - *Shall I wait? : Please wait. = Shall I call a taxi? : Please call a taxi.*
     No solution : ちょっと待ってください。 = タクシーをお呼びしますか。 : タクシーを呼んでください。
   - *Shall I wait? : Would you wait for me? = Shall I call a taxi? : Would you call a taxi for me?*
     No solution : ここでちょっと待っていてください。 = タクシーをお呼びしますか。 : タクシーを呼んでいただけませんか。
   - *Shall I wait? : Please wait. = Shall I call an ambulance? : Please call an ambulance.*
     No solution : ちょっと待ってください。 = 救急車を呼びましょうか。 : 救急車を呼んでください。
   - *Shall I wait? : Please wait. = Shall I call a doctor? : Please call a doctor.*
     No solution : ちょっと待ってください。 = 医者をお呼びしましょうか。 : 医者をよんでください。
   - *Shall I wait? : Would you wait for me? = Shall I call a doctor? : Would you call a doctor for me?*
     No solution : ここでちょっと待っていてください。 = 医者をお呼びしましょうか。 : 医者を呼んでもらえますか。
   - *Shall I wait? : Shall I hold it? = Could you wait for me? : Could you hold it for me?*
     No solution : では、お取り置きしておきましょうか。 = 待ってもらえませんか。 : これ、ちょっと持ってくれる。
   - *Shall I wait? : Shall I wrap it? = Could you wait? : Could you wrap it?*
     No solution : お包みしましょうか。 = しばらくお待ちください。 : 包装してください。
   - *Shall I wait? : Shall I wrap it? = Could you wait for me, please? : Could you wrap it for me, please?*
     No solution : お包みしましょうか。 = ちょっと待っていてもらえますか。 : 包んでいただけますか。

- *Shall I wait? : Could you wait? = Shall I call a taxi? : Could you call a taxi?*
  No solution : しばらくお待ちください。＝タクシーをお呼びしますか。：タクシーを呼んで下さい。
- *Shall I wait? : Could you wait for me? = Shall I call a taxi? : Could you call a taxi for me?*
  No solution : 待ってもらえませんか。＝タクシーをお呼びしますか。：タクシーを呼んでいただけますか。
- *Shall I wait? : Would you wait for me, please? = Shall I call a taxi? : Would you call a taxi for me, please?*
  No solution : 待っていただけますか。＝タクシーをお呼びしますか。：タクシーを呼んでもらえますか。
- *Shall I wait? : Could you wait? = Shall I call an ambulance? : Could you call an ambulance?*
  No solution : しばらくお待ちください。＝救急車を呼びましょうか。：救急車を呼んでもらえますか。
- *Shall I wait? : Could you wait? = Shall I call a doctor? : Could you call a doctor?*
  No solution : しばらくお待ちください。＝医者をお呼びしましょうか。：往診を頼んでいただけますか。
- *Shall I wait? : Could you wait for me? = Shall I call a doctor? : Could you call a doctor for me?*
  No solution : 待ってもらえませんか。＝医者をお呼びしましょうか。：医者を呼んでもらえますか。
- *Shall I wait? : Would you wait for me, please? = Shall I call an ambulance? : Would you call an ambulance for me, please?*

  No solution : 待っていただけますか。＝救急車を呼びましょうか。：救急車を呼んでください。
- *Shall I wait? : Could you wait? = Shall I check the oil and water? : Could you check the oil and water?*
  No solution : しばらくお待ちください。＝オイルや水をチェックしましょうか。：オイルとラジエータを点検してください。
- *Shall I wait? : Shall I call a taxi? = Could you wait for me, please? : Could you call a taxi for me, please?*
  No solution : タクシーをお呼びしますか。＝ちょっと待っていてもらえますか。：タクシーを呼んでいただけませんか。
- *Shall I wait? : Could you wait for me, please? = Shall I call an ambulance? : Could you call an ambulance for me, please?*

  No solution : ちょっと待っていてもらえますか。＝救急車を呼びましょうか。：救急車を呼んでいただけませんか。
- *Shall I wait? : Could you wait for me, please? = Shall I call a doctor? : Could you call a doctor for me, please?*

  No solution : ちょっと待っていてもらえますか。＝医者をお呼びしましょうか。：医者を呼んでもらえますか。

3. Can I have your ticket, please?:

- *Can I have your ticket, please? : May I have the bill, please? = Can I see your ticket, please? : May I see the bill, please?*

  No solution : お勘定をお願いします。＝切符を拝見できますか。：ちょっと明細書を見せてもらえますか。
- *Can I have your ticket, please? : May I have the menu, please? = Can I see your ticket, please? : May I see the menu, pleas*

  No solution : メニューをいただけますか。＝切符を拝見できますか。：メニューを見せてください。
- *Can I have your ticket, please? : May I have your claim tag? = Can I see your ticket, please? : May I see your claim tag?*

  No solution : 荷物引換証をいただけますか。＝切符を拝見できますか。：預り証を見せていただけますか。
- *Can I have your ticket, please? : Can I have the menu, please? = May I see your ticket, please. : May I see the menu, please*

  No solution : メニューを見せてください。＝切符を見せてください。：メニューを見せてもらえますか。
- *Can I have your ticket, please? : Can I have the bill, please? = May I see your ticket, please? : May I see the bill, please?*

  No solution : 精算書をいただけますか。＝切符を拝見します。：ちょっと明細書を見せてもらえますか。
- *Can I have your ticket, please? : Can I have the menu, please? = May I see your ticket, please? : May I see the menu, pleas*

  切符を拝見します。：メニューを見せてください。＝切符を拝見します。：メニューを見せてください。
- *Can I have your ticket, please? : Can I have the bill, please? = Can I see your ticket, please? : Can I see the bill, please?*

  No solution : 精算書をいただけますか。＝切符を拝見できますか。：明細を確認させてください。
- *Can I have your ticket, please? : Can I have a menu, please? = Can I see your ticket, please? : Can I see a menu, please?*

  No solution : メニューはありますか。＝切符を拝見できますか。：メニューをください。
- *Can I have your ticket, please? : Could I have the bill, please? = Can I see your ticket, please? : Could I see the bill, please?*

  No solution : すみません、お勘定をお願いします。＝切符を拝見できますか。：明細書を見せていただけますか。
- **Can I have your ticket, please?** : *Can I have the menu, please? = Can I see your ticket, please? : Can I see the menu, p*

  切符を拝見できますか。：メニューを見せてください。＝切符を拝見できますか。：メニューを見せてください。

- *Can I have your ticket, please? : Can I have another one, please? = Can I see your ticket, please? : Can I see another one,*

  No solution : もう一杯ください。 = 切符を拝見できますか。 : 他のものを見ることができますか。

- *Can I have your ticket, please? : May I have the wine list, please? = Can I see your ticket, please? : May I see the wine list,*

  No solution : ワインリストを見せていただけますか。 = 切符を拝見できますか。 : ワインリストを見せてください。

- *Can I have your ticket, please? : Can I have this, please? = May I see your ticket, please? : May I see this, please?*

  No solution : これを下さい。 = 切符を拝見します。 : これを、見せてください。

- *Can I have your ticket, please? : May I have your passport, please? = Can I see your ticket? : May I see your passport?*

  切符を見せてください。 : パスポートを見せてください。 = 切符を拝見します。 : パスポートを拝見します。

- *Can I have your ticket, please? : Can I have this, please? = Can I see your ticket, please? : Can I see this, please?*

  No solution : これを下さい。 = 切符を拝見できますか。 : これを見せてもらえますか。

- *Can I have your ticket, please? : May I have your passport, please? = Can I see your ticket, please? : May I see your passpo*

  No solution : パスポートを見せてください。 = 切符を拝見できますか。 : パスポートを見せていただけますか。

参考文献

[CSTAR 03] `http://c-star.atr.co.jp/cstar-corpus/corpus.htm`

[Itkonen & Haukioja 97] Itkonen, E. and Haukioja, J.
A rehabilitation of analogy in syntax (and elsewhere)
in András Kertész (ed.) Metalinguistik im Wandel: die kognitive Wende in Wissenschaftstheorie und Linguistik. Frankfurt a/M, Peter Lang, (1997) 131-177.

[Lepage 92] Yves Lepage
Easier C programming – Some useful objects
ATR report TR-I-0294, Kyoto, November 1992.

[Lepage 01] Yves LEPAGE
Analogy and formal languages
Proceedings of FG/MOL 2001, Helsinki, 2001.
`http://www.elsevier.nl/locate/entcs/volume47.html`.

[Lepage 03] Yves Lepage
De l'analogie rendant compte de la commutation en linguistique
Mémoire d'habilitation à diriger les recherches, 2003.
`http://www.slt.atr.co.jp/l̃epage/pdf/dhdryl.pdf.gz`