NatServer: A Client-Server Architecture for building Parallel Corpora applications

Alberto Simões
Dept. Informática – Univ. do Minho
ambs@di.uminho.pt

José João Almeida
Dept. Informática – Univ. do Minho
jj@di.uminho.pt

Resumen: Los corpora paralelos son importantes para la mayoría de las tareas de procesamiento de Lenguaje Natural. Gran parte de los investigadores utilizan los corpora paralelos empleando-los en aplicaciones comunes, como la traducción máquina, y en tareas mono-lingüísticas, como sean la detección de paráfrasis y la resolución de problemas de ambigüedad semántica. Este trabajo ha sido orientado por estas cuestiones y propone una arquitectura cliente-servidor de interrogación eficiente de corpora paralelos y diccionarios de traducción probabilística.

Palabras clave: corpora paralelos, indexación de corpora

Abstract: Parallel corpora are important resources for most Natural Language processing tasks. From the common applications, like machine translation, to the usually mono-lingual tasks as paraphrase detection and word sense disambiguation, most researchers are using massive parallel corpora. Thus, the availability of an efficient way to manage them is very important. This paper presents a Client-Server architecture to query efficiently parallel corpora and probabilistic translation dictionaries.

Keywords: parallel corpora, corpora indexing

1 Motivation

Parallel Corpora are being used for a large range of applications in Natural Language Processing. Nowadays, most typically monolingual research area are benefiting from parallel corpora usage.

To help working with big amounts of parallel corpora we propose a client-server approach to store them, and make it accessible for querying.

When developing NATools we had the following points in mind:

- be able to query for concordancies, both simple and parallel: search for a word or pattern, in the source or target language, or in both at the same time;
- be able to query Probabilistic Translation Dictionaries[1]. This way we can relate words from corpora translation units.
- be able to query more than one corpus at the same time, and with different languages. Also, to be able to query for corpora meta-information like involved languages, number of translation units and other;
- support big corpus, more than one million translation unit. Most studies are statistics, and the results precision highly depends on the corpus size.
- fast for interactive and batch tasks, which lead us to a double architecture:
  - reduce loading time for indexes and dictionaries when using them interactively, like in a web-based application (we do not want the user to wait a long time for the answer or that the web server timeouts). With this in mind, there is a server which loads all the information just once reducing the load time.
  - reduce the overhead time for the communication. For batch processes which query repeatedly the same corpus, it is better to load the corpus indexes and query them in memory. In these cases, the load time overhead is too small compared with the overall time of the process.
- easy to distribute work:
  - for big corpora we can split them in small chunks and make them available for querying from different servers.
  - different applications or users be able to query the same corpus in the same server, thus reducing the need for replication. For instance, for the implementation of distributed translation memories (Simões, Guinovart, and Almeida, 2004).
- be an open-source and free tool. While there are some applications to manage corpora (like Corpus Query Processor (König, 1999) or SARA for the BNC corpus (Dodd, 1997)) they are not freely available. SARA is a commercial software and CQP is just available for research with license limitations. There are some

[1] Next section will detail what PTDs are, and how we obtain them.

web-based tools like TransSearch (RALI Laboratory, 2006) and COMPARA (Frankenberg-Garcia and Santos, 2003) to query corpora. The first is paid. COMPARA is freely available but uses as backend CQP. To be an open-source tool is important for other researchers be able to enhance the program accordingly with their needs, to compare times, and other.

For special situations to use a general relational database engine might be sufficient. For instance, in (Sarmento, 2006) the mySQL database engine is used with success but for the specific n-gram based queries.

- prepare a simple API to write server clients with few lines;
- develop a set of real web-clients to test and validate the tool, and to test the API;
- develop multilayer support: add levels of information related to corpora words. The base layer includes the words or tokens that constitute the corpus. Other layers add information like lemmatization or part-of-speech.

2 *NATools*

This section presents a brief overview of how NATools (Hiemstra, 1996; Hiemstra, 1998; Simões, 2004; Simões and Almeida, 2003) works, namely the corpus encoder, and the extractor of Probabilistic Translation Dictionaries (PTD).

2.1 NATools Pipeline

NATools is composed of different modules, which work as a pipeline:

Corpora Encoder:

- a corpus splitter: divides the corpus in smaller chunks that can be aligned independently in memory. This is important because it makes the tools scalable for big corpora[2].
- a corpus encoder: maps an integer identifier to each word. Then, chunks are codified to use these integer identifiers. This step also creates indexes used by NatServer to query the corpora efficiently. If the objective is just corpus query (and not probabilistic translation dictionaries) just these first two steps are really needed.

  This encoding method let us work with corpora layers (used by (Petersen, 2004) and also discussed in (Graça, 2006)). The described method creates the first layer containing just the words (each one has an unique identifier). Other levels can be added relating information to each word (or set of words) like lemmatization or part-of-speech tagging.

PTD Builder:

- a co-occurrence counter: creates a matrix of co-occurrences, where each cell of the matrix contains the number of times two words occurred on the same translation unit.
- the EM-Algorithm: the Expectation-Maximization algorithm iterates over the matrix of co-occurrences finding maximum likelihood estimates.
- dictionary creation: interprets the co-occurrences matrix and dumps a pair of dictionary files.
- join dictionaries: the four previous step run over each corpus chunk. Thus, when finishing, we need to join the dictionary files (merge and weight them accordingly with chunk sizes).

The result of the full process is the encoded corpora and query indexes and a pair of probabilistic translation dictionaries.

2.2 Probabilistic Translation Dictionaries

Probabilistic Translation Dictionaries have this structure:

$$w_\alpha \rightharpoonup (occur \times w_\beta \rightharpoonup P\left(\mathcal{T}\left(w_\alpha\right) = w_\beta\right))$$

That is, a map from each word on the source language ($\mathcal{L}_\alpha$) to a pair: the number of occurrences of that word in the corpus, and a map from possible translations words to its respective probability of being a translation.

Note that PTDs are not traditional dictionaries. They contain information about strong relationship between words in two different languages (where most of them are translations).

The following extract is from EuroParl (Koehn, 2002). Europarl is more than a million translation units in size, about 30 million words in each language. The resulting PTD include about 100 000 entries, each with 1 to 8 possible translations.

```
** Word: europe
** OccurrenceCount: 42853

      europa: 94.71 %
    europeus:  3.39 %
     europeu:  0.81 %
    europeia:  0.11 %

** Word: stupid
** OccurrenceCount: 180

    estúpido:  17.55 %
    estúpida:  10.99 %
   estúpidos:   7.41 %
```

[2] The examples we show in this article are from EuroParl (Koehn, 2002), a corpus with a million translation units. Its alignment process creates 15 chunks of corpora aligned independently.

```
      avisada:   5.65 %
      direita:   5.58 %
      impasse:   4.48 %
      ocupado:   3.75 %
```

3 Server Architecture

NATools is a basic socket server. It accepts requests in a specific port and sends answers.

During the encode and alignment process described above, a directory with the encoded corpora, search indexes and dictionaries will be created on the hard disk. This directory includes all files needed by NatServer to query the corpus and the probabilistic translation dictionary.

The server is configured with a text file with directory paths. These are (absolute) paths to corpora encoded with NATools. The server will load the main lexicon tables and the probabilistic translation dictionaries to memory. The socket is opened in the system and the server is ready to answer the clients.

During operation NatServer will load the indexes and cache corpora chunks while they are needed, but will not keep them in memory.

Given that the server supports more than one corpus at the same time, and supports both concordance or probabilistic translation dictionaries querying, it was necessary to build an interface to obtain information about the meta-data of corpora currently available.

There follows the description of the server API for querying meta-information, probabilistic translation dictionaries and concordances[3].

3.1 Meta-Information Query

The first query normally issed by the client asks for the list of available corpora on the server. It will return a list of names as well as the repective identifiers and involved languages:

$$LIST : \xi \longrightarrow (id \times name \times \mathcal{L}_\alpha \times \mathcal{L}_\beta)^\star$$

The corpus identifier will be required for all other queries. For instance, To query meta-information, you supply the corpus identifier and the meta-data attribute you are interested on (for example, name, description, number of translation units and others):

$$QUERYATTR : id \times attribute \longrightarrow varvalue$$

This query will return the empty string if the attribute does not exist. Attributes are not confined to the ones created by NATools, as you can add your own to each corpus configuration file.

3.2 Probabilistic Translation Dictionary Query

For each parallel corpus there is a pair of probabilistic translation dictionaries: for two languages $\mathcal{L}_\alpha$ and $\mathcal{L}_\beta$ there are $\mathcal{L}_\alpha \rightarrow \mathcal{L}_\beta$ and $\mathcal{L}_\alpha \leftarrow \mathcal{L}_\beta$. So, when querying it, you should supply not only the corpus identifier and the word you are searching for, but also the *direction* (of the dictionary) you are using.

The answer from the server is the number of occurrences of that word in the corpus, and a list of possible translations and their probability:

$$\begin{gathered} QUERYPTD : id \times word_{\mathcal{L}_\alpha} \times direction \\ \downarrow \\ occurrences \times word\mathcal{L}_\beta \mapsto probability \end{gathered}$$

3.3 Concordance Query

There are two distinct concordance queries: you can search for translation units where a set of words appear in any order, or for translation units where a set of word appear sequentially in a specific order. We call the first "Word concordance" and the second "Pattern concordance".

Both query methods can be done searching just one of the languages or both at the same time. The two methods signatures are the same, as the only real difference relies on the semantic the server applies to each method. Also, when asking for "Pattern concordance" you can supply a place-holder (the special word "asterisk" — *) which represent a placeholder for any word.

$$\begin{gathered} CONC : id \times \left(word^\star_{\mathcal{L}_\alpha} \ast word^\star_{\mathcal{L}_\beta}\right) \\ \downarrow \\ \left(sentence_{\mathcal{L}_\alpha} \times sentence_{\mathcal{L}_\beta}\right)^\star \end{gathered}$$

4 Clients

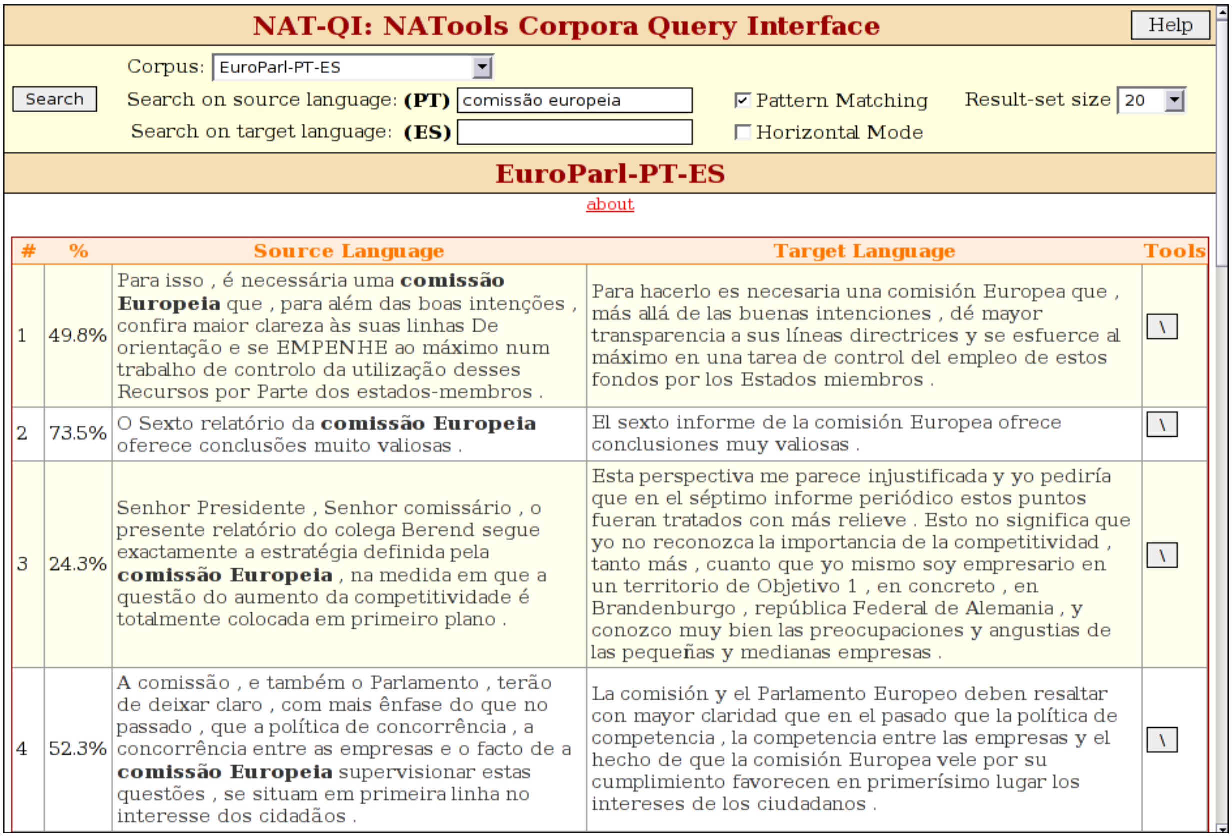
In order to make the Natural Language Resources stored with NatServer useful not just for our research tools, we built a set of web clients being used for linguists and students to query and navigate through corpora (section 4.2). Regarding our research, we are using some more complex tools that use NatServer to obtain information for extraction of sub-sentence alignments (section 4.3).

All these clients are available in NATools package, and are easily installable and configurable.

4.1 Command-line Concordance Tool Example

To illustrate the way the Perl API simplifies the process of writing clients for NatServer, in this section we show a little Perl program to query for translation units with a set of words.

[3] On (Simões and Almeida, 2003) similar web tools for concordancies and probabilistic translation dictionary were presented. They were redesigned to use NatServer and to support more than one corpus at a time.



NAT-QI: NATools Corpora Query Interface

Corpus: EuroParl-PT-ES

Search on source language: **(PT)** comissão europeia

☑ Pattern Matching

Result-set size 20

Search on target language: **(ES)**

□ Horizontal Mode

EuroParl-PT-ES

about

#	%	Source Language	Target Language	Tools
1	49.8%	Para isso , é necessária uma **comissão Europeia** que , para além das boas intenções , confira maior clareza às suas linhas De orientação e se EMPENHE ao máximo num trabalho de controlo da utilização desses Recursos por Parte dos estados-membros .	Para hacerlo es necesaria una comisión Europea que , más allá de las buenas intenciones , dé mayor transparencia a sus líneas directrices y se esfuerce al máximo en una tarea de control del empleo de estos fondos por los Estados miembros .	\
2	73.5%	O Sexto relatório da **comissão Europeia** oferece conclusões muito valiosas .	El sexto informe de la comisión Europea ofrece conclusiones muy valiosas .	\
3	24.3%	Senhor Presidente , Senhor comissário , o presente relatório do colega Berend segue exactamente a estratégia definida pela **comissão Europeia** , na medida em que a questão do aumento da competitividade é totalmente colocada em primeiro plano .	Esta perspectiva me parece injustificada y yo pediría que en el séptimo informe periódico estos puntos fueran tratados con más relieve . Esto no significa que yo no reconozca la importancia de la competitividad , tanto más , cuanto que yo mismo soy empresario en un territorio de Objetivo 1 , en concreto , en Brandenburgo , república Federal de Alemania , y conozco muy bien las preocupaciones y angustias de las pequeñas y medianas empresas .	\
4	52.3%	A comissão , e também o Parlamento , terão de deixar claro , com mais ênfase do que no passado , que a política de concorrência , a concorrência entre as empresas e o facto de a **comissão Europeia** supervisionar estas questões , se situam em primeira linha no interesse dos cidadãos .	La comisión y el Parlamento Europeo deben resaltar con mayor claridad que en el pasado que la política de competencia , la competencia entre las empresas y el hecho de que la comisión Europea vele por su cumplimiento favorecen en primerísimo lugar los intereses de los ciudadanos .	\

Figure 1: NatSearch: do corpora concordances.

```
use NAT::Client;
$server = NAT::Client->new(
   PeerAddr=>'eremita.di.uminho.pt');
$concs = $server->conc({crp=>1},
                join(" ",@ARGV));
for my $tu (@$concs) {
  print "$tu->[0]\n";
  print "$tu->[1]\n";
  print "\n"
}
```

The first line imports the NAT::Client module, which includes the API for querying NatServer. The second line creates a new client. Here we can specify the remote host where the queries will be send, or the local path for a local corpora we want to query. In this case, we change the client creation code to:

```
$server = NAT::Client->new(
   local => '/corpora/EuroParl-PT-EN');
```

There follows the line with the query. The first argument is a configuration hash where we can describe the corpus to query. Follows the string to be searched (the arguments to the script). The `for` loop iterates over the translation units where the words occur, and print them.

This sample code can be used from the command line as:

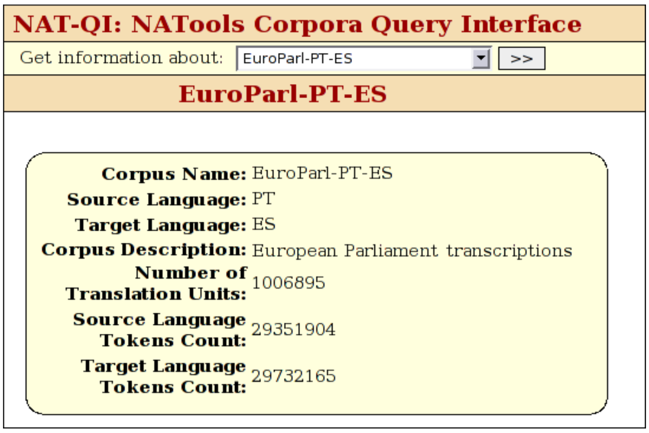
```
$ perl example parlamento europeu
```

and the result will be something like:

```
  Declaro reaberta a sessão do Parlamento Europeu ,
que tinha sido interrompida na sexta-feira , 17 de
Dezembro último , e renovo todos os meus votos ,
esperando que tenham tido boas férias .
  Declaro reanudado el período de sesiones del
Parlamento Europeo , interrumpido el viernes 17 de
diciembre pasado, y reitero a Sus señorías mi deseo
de que hayan tenido unas buenas vacaciones .

  Uma das pessoas recentemente assassinadas foi o
senhor Kumar Ponnambalam, que ainda há poucos meses
visitara o Parlamento Europeu
  Una de las personas que recientemente han
asesinado en Sri Lanka ha sido al Sr . Kumar
Ponnambalam , quien hace pocos meses visitó el
Parlamento Europeo .

  Senhora Presidente , coincidindo com a primeira
sessão deste ano do Parlamento Europeu, nos Estados
Unidos , no Texas , está marcada , lamentavelmente
para a próxima quinta-feira , a execução de um
condenado à morte , um jovem de 34 anos a quem
designaremos por X .
  Señora Presidenta , coincidiendo con el primer
período parcial de sesiones de este año del
Parlamento Europeo , lamentablemente , en los
Estados Unidos , en Texas , se ha fijado para el
próximo jueves la ejecución de un condenado a la
pena capital , un joven de 34 años que llamaremos
con el nombre de Hicks .
```



NAT-QI: NATools Corpora Query Interface

Get information about: EuroParl-PT-ES >>

EuroParl-PT-ES

Corpus Name:	EuroParl-PT-ES
Source Language:	PT
Target Language:	ES
Corpus Description:	European Parliament transcriptions
Number of Translation Units:	1006895
Source Language Tokens Count:	29351904
Target Language Tokens Count:	29732165

Figure 2: NatAbout: query corpora meta-information.

4.2 Basic Web Clients

As a first set of clients we created three web applications to access encoded corpora. Figure 2 shows the meta-information CGI where the user can select a corpus and see the respective languages, name, description and measures.

The concordance web interface is similar to other parallel corpora tools. Figure 1 shows a query and the result. Note the second column with a quality measure of the translation unit, which was pre-calculated using information from the PTDs.

The probabilistic translation dictionary web interface is shown in figure 3. It shows two levels of the dictionary in a compact table. It is possible to change the view format to an expanded version, as well as to follow word links and check translation units where a specific pair of words occur.

4.3 Tool Clients

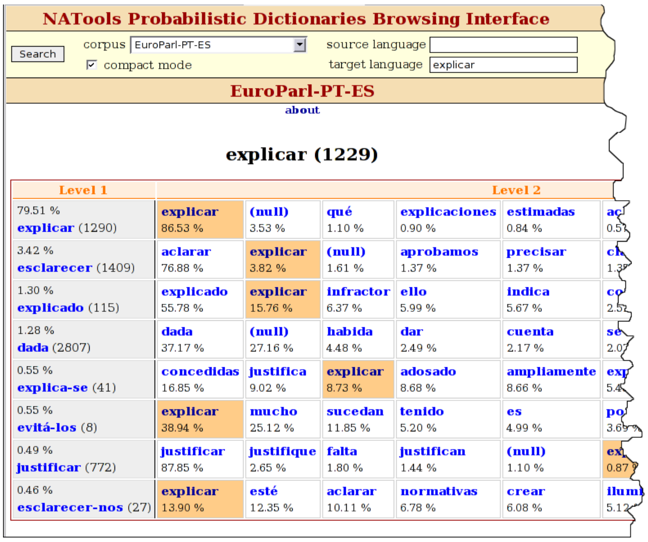
While it is interesting to query the server using a CGI that is not one of the most important factors of using it. In fact, the server is very useful for batch processes where corpora and/or the dictionaries need to be queried.

4.3.1 Combinatory examples extraction

One of the purposes for the server is the development of a example based machine translation system prototype. As the example extraction algorithm(Simões and Almeida, 2006) does not run fast enough to be used on-the-fly, we are caching examples using a batch processor.

To extract these examples, we query the server for translation units. For each word in this translation unit we query the server for its probabilistic translation dictionary. This means that a single translation unit with ten words in each language does twenty-two connections to the Nat-Server.

This is a task where one might argue that the socket connection is an overhead in the process, and that if we load the corpus information just once, the loading time will be insignificant compared to the time the full task will take. With that in mind, our client API lets the user load the full corpus to memory and use it directly, instead of connecting to a server.



NATools Probabilistic Dictionaries Browsing Interface

corpus EuroParl-PT-ES source language

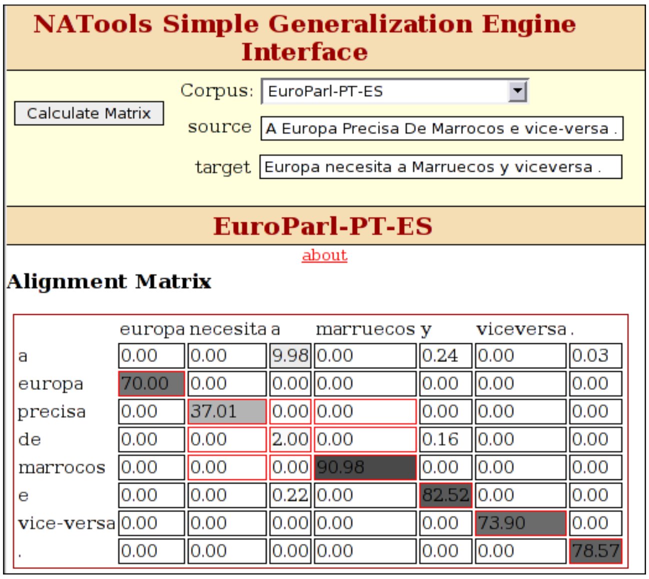
☑ compact mode target language explicar

EuroParl-PT-ES

explicar (1229)

Level 1	Level 2					
79.51 % **explicar** (1290)	**explicar** 86.53 %	**(null)** 3.53 %	**qué** 1.10 %	**explicaciones** 0.90 %	**estimadas** 0.84 %	**ac** 0.57
3.42 % **esclarecer** (1409)	**aclarar** 76.88 %	**explicar** 3.82 %	**(null)** 1.61 %	**aprobamos** 1.37 %	**precisar** 1.37 %	**cl** 1.37
1.30 % **explicado** (115)	**explicado** 55.78 %	**explicar** 15.76 %	**infractor** 6.37 %	**ello** 5.99 %	**indica** 5.67 %	**co** 2.57
1.28 % **dada** (2807)	**dada** 37.17 %	**(null)** 27.16 %	**habida** 4.48 %	**dar** 2.49 %	**cuenta** 2.17 %	**se** 2.07
0.55 % **explica-se** (41)	**concedidas** 16.85 %	**justifica** 9.02 %	**explicar** 8.73 %	**adosado** 8.68 %	**ampliamente** 8.66 %	**exp** 5.4
0.55 % **evitá-los** (8)	**explicar** 38.94 %	**mucho** 25.12 %	**sucedan** 11.85 %	**tenido** 5.20 %	**es** 4.99 %	**po** 3.69 %
0.49 % **justificar** (772)	**justificar** 87.85 %	**justifique** 2.65 %	**falta** 1.80 %	**justifican** 1.44 %	**(null)** 1.10 %	**ex** 0.87
0.46 % **esclarecer-nos** (27)	**explicar** 13.90 %	**esté** 12.35 %	**aclarar** 10.11 %	**normativas** 6.78 %	**crear** 6.08 %	**ilumi** 5.12

Figure 3: NatDict: search probabilistic translation dictionaries.



NATools Simple Generalization Engine Interface

Corpus: EuroParl-PT-ES

source A Europa Precisa De Marrocos e vice-versa .

target Europa necesita a Marruecos y viceversa .

EuroParl-PT-ES

Alignment Matrix

	europa	necesita	a	marruecos	y	viceversa	.
a	0.00	0.00	9.98	0.00	0.24	0.00	0.03
europa	70.00	0.00	0.00	0.00	0.00	0.00	0.00
precisa	0.00	37.01	0.00	0.00	0.00	0.00	0.00
de	0.00	0.00	2.00	0.00	0.16	0.00	0.00
marrocos	0.00	0.00	0.00	90.98	0.00	0.00	0.00
e	0.00	0.00	0.22	0.00	82.52	0.00	0.00
vice-versa	0.00	0.00	0.00	0.00	0.00	73.90	0.00
.	0.00	0.00	0.00	0.00	0.00	0.00	78.57

Figure 4: Examples extraction based on probabilistic translation dictionaries.

4.3.2 Web example extraction tool

The algorithm for the example extraction uses a matrix of probability values, and tries to find cells corresponding to true word translations. While tuning and debuging the algorithm we needed to see how it was evolving. For that purpose we developed a CGI to show graphically the compued matrix. Figure 4 shows an example of that CGI in action.

5 *Measures*

Our test environment is not the most favorable: we are running both the server and the client in the same machine, which leads to overheads during process switching.

Although the server does not have a fixed limit for number of corpora and its size, it all depends on the server hardware being used.

We normally work with about four to five corpora loaded at the same time. For instance, the tests related before were done with four corpora loaded (EuroParl PT/ES, PT/EN and PT/FR, each with more than one million translation units and a small four thousand translation unit test corpus PT/LA). Each EuroParl corpora takes about 300MB of ASCII text. The server uses about 600 MegaBytes of RAM memory with these four corpora loaded, included respective PTDs. The hardware used for our tests is an Intel Pentium 4, 3GHz with 2MB of RAM running Linux.

We did a set of 100 000 requests to the server for the first 20 concordancies (both word and pattern concordancies) to calculate the medium time needed to answer a request, and the number of requests answered per second. Table 1 resumes these tests times.

		sec/req	req/sec	occs
1	cão	0.038	26.027	40
2	europa	0.010	98.090	36532
3	parlamento europeu	0.036	27.131	23841
4	“parlamento europeu”	0.036	27.485	23841
5	“europeu parlamento”	1.474	0.68	23841
6	PTD(parlamento)	0.001	1676.45	–

Table 1: NatServer timings — 1, 2 and 3 are times for word concordancies, 4 and 5 are times for pattern concordancies and 6 is the timing for PTD query.

Tests 1 and 2 are basically the same, just changing the word being searched. Because “cão” appears just 40 times in the corpus, it should be distributed in the different encoded chunks of the corpus. Thus, a lot more files will be opened to retrieve the occurrencies. As “europa” appears a lot more, the first 20 occurrences are probably in the first encoded chunk.

Tests 3 and 4 show that the pattern matching algorithm is not adding time to the results. Basically, everytime “parlamento” and “europeu” appear in the same sentence they normally appear as “parlamento europeu”, so, the work to get the first 20 occurrences in word or pattern matching is basically the same.

Finally, test 5 is a unfavorable test. There are no occurrences of “europeu parlamento” in the corpus but there are a 23841 sentences where these two words occur. This means that the server will retrieve those 23841 sentences. This result shows that work is needed in bigrams indexing.

Regarding loading time the server takes about 4 seconds to load the indexes for these four corpora.

Table 2 summarizes the number of PTD queries NatServer can answer by second. the server can answer more than 1700 PTDs queries per second. In case we need batch processes like the example extraction we referred before, we can load the corpus directly in the main program and we get 45455 queries answered by second. While this seems a lot better than the client/server architecture we need to stress that if we load the corpus data for every request (what a CGI would do), the system would be able to answer 0.7 requests per second!

	req/sec
Server queries	1 737.92
Local queries	45 454.55
Local queries with load	0.70

Table 2: Number of requests answered by second.

6 Conclusions

While the server is not yet as powerful and efficient as other tools, namely CQP, it is very flexive and performance is evolving each day. Our first objective is to use the server for example based machine translation (Somers, 1999) tools. Thus, exatracted examples will be served as if they were normal corpora. Also, more types of information will be added to the server:

- a shared probabilistic translation dictionary, result of adding up all available PTDs (not just the ones being served). This dictionary, being the result of analyzing much more corpora should be more accurate than the separated ones;
- some of the tools being developed for example extraction use morphological analyzers(Almeida and Pinto, 1994) for the involved languages. Work is done on analyzing how the performance would increase if this information is added to the server;

As referred before, NATools corpora can have more than one layer of information. At the moment the server is just querying the base level (words and sentences). More work is being done to make NatServer aware of the available layers.

The server can be tested on `http://eremita.di.uminho.pt/albin/nat` and the source code can be downloaded from `http://natools.sf.net/`.

References

Almeida, J. João and Ulisses Pinto. 1994. Jspell — um módulo para análise léxica genérica de linguagem natural. In *Actas do Congresso da Associação Portuguesa de Linguística*.

Dodd, Tony. 1997. Sara: Technical manual-release 930. `http://www.natcorp.ox.ac.uk/sara/TechMan/index.htm`, November.

Frankenberg-Garcia, Ana and Diana Santos, 2003. *Introducing COMPARA, the Portuguese-English parallel translation corpus*, pages 71–87. St. Jerome Publishing, Manchester.

Graça, João de Almeida Varelas. 2006. A framework for integrating natural language tools. Master's thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, February.

Hiemstra, Djoerd. 1996. Using statistical methods to create a bilingual dictionary. Master's thesis, Department of Computer Science, University of Twente, August.

Hiemstra, Djoerd. 1998. Multilingual domain modeling in twenty-one: automatic creation of a bi-directional lexicon from a parallel corpus. Technical report, University of Twente, Parlevink Group.

Koehn, Philipp. 2002. Europarl: A multilingual corpus for evaluation of machine translation. Draft, Unpublished, `http://people.csail.mit.edu/~koehn/publications/europarl.ps`.

König, Oliver Christ & Bruno M. Schulze & Anja Hofmann & Esther. 1999. *The IMS Corpus Workbench: Corpus Query Processor (CQP): User's Manual*. Institute for Natural Language Processing, University of Stutgart.

Petersen, Ulrik. 2004. Emdros — a text database engine for analyzed or annotated text. In *Coling 2004*, Geneva.

RALI Laboratory. 2006. TransSearch. `http://www.tsrali.com`.

Sarmento, Luís. 2006. BACO — a large database of text and co-occurrences. In *5th International Conference on Language Resources and Evaluation (LREC'2006)*, Genova, May.

Simões, Alberto and J. João Almeida. 2006. Combinatory examples extraction for machine translation. In *EAMT 11th Annual Conference*, Oslo, Norway, June.

Simões, Alberto, Xavier Gómez Guinovart, and José João Almeida. 2004. Distributed translation memories implementation using webservices. pages 89–94. Sociedade Española para el Procesamiento del Lenguaje Natural, Jul.

Simões, Alberto M. and J. João Almeida. 2003. Natools – a statistical word aligner workbench. *SEPLN*, Sep.

Simões, Alberto Manuel Brandão. 2004. Parallel corpora word alignment and applications. Master's thesis, Escola de Engenharia - Universidade do Minho.

Somers, Harold. 1999. Review article: Example based machine translation. *Machine Translation*, 14(2):113–157.