NATools – A Statistical Word Aligner Workbench

Alberto Manuel Simões
Departamento de Informática
Universidade do Minho
albie@alfarrabio.di.uminho.pt

José João Almeida
Departamento de Informática
Universidade do Minho
jj@di.uminho.pt

Resumen: Este documento presenta el proyecto TerminUM y el trabajo realizado en su alineador estadístico a nivel de palabra (NATools). Muestra una variedad de métodos de alineamento para corpora paralelos y discute los diccionarios terminológicos resultantes y su uso: evaluación de traducciones; construcción de un sistema de navegación para estudios lingüísticos, o traducción estadística.
Palabras clave: corpora paralelos, alineamento a nivel de palabra

Abstract: This document presents the TerminUM project and the work done in its statistical word aligner workbench (NATools). It shows a variety of alignment methods for parallel corpora and discusses the resulting terminological dictionaries and their use: evaluation of sentence translations; construction of a multi-level navigation system for linguistic studies or statistical translations.
Keywords: parallel corpora, word alignment

1 Introduction: the TerminUM Project

The TerminUM project aims at the development of tools to produce multilingual resources, and free resources disponibilization. To formalize the project structure, a graph is defined, where data types and processes between them are formalized as shown in figure 1. We think that each node (data-type) is a deliverable that should be made available by the project. The edges are process transformations which correspond to different tools. Each one of these processes is a research task: to design, validate and improve the tool.

This life-cycle begins by using parallel texts we have in our system, or detecting parallel web-sites in the Internet: in article "Grabbing parallel corpora from the web" (Almeida, Simões, and Castro, 2002) we present a detailed tour over the parallel web-site detection and extraction. In section 1.1 we give a short summary of these methods for completeness.

Having these candidate pairs in our system — say a sequence of pairs: $(\text{File}^2)^\star$ — we need to validate them, comparing file sizes, non-textual content and using some other heuristics to determine if we can consider this candidate pair a true pair of parallel texts.

The next step is to divide the files into sentences — a pair of sentence sequences (with an optional file identifier)

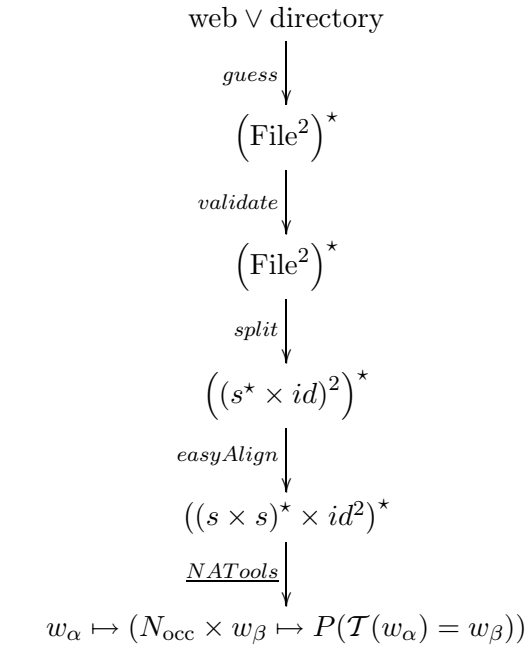

Figure 1: TerminUM corpora life cycle

$(s^\star \times id)^2$ — which will be pre-processed with Perl scripts and aligned with CWB Corpus Workbench(König, 1999) *easy-align* or using a vanilla aligner(Gale and Church, 1991; Danielsson and Ridings, 1997):

1. remove tags or commands specific from file formats;

2. use a NLP tool to detect sentences boundaries;
3. create an XML file for each language with synchronization points;
4. align at sentence level;

This produces a list of aligned sentences — $((s \times s)^\star \times id)^\star$.

Finally, we use the parallel corpus to create Translation Memory Exchange (OSCAR, 2003; Savourel, 1997) files and vice-versa. The word alignment is done using a variant of Twente-aligner(Hiemstra, 1998; Hiemstra, August 1996) which produces translation dictionaries:

$$w_\alpha \mapsto (N_{\text{occ}} \times w_\beta \mapsto P(\mathcal{T}(w_\alpha) = w_\beta))$$

This structure is detailed on section 2.2. These dictionaries can be used for different tasks as presented on section 3.

1.1 Grabbing Parallel Corpora from the Web

To grab parallel texts from the web we need to find candidate pairs and to validate them. To detect candidate pairs we use four distinct techniques. The first one is based on (Resnik, 1998; Resnik, 1999) queries to web search engines.

The second one, which is giving better results at the moment, uses heuristics over URLs paths. This method is based on the natural organization of files: Webmasters who needs to publish a web-site on multiple languages begins organizing this information under directories with the language name or using prefixes or suffixes in the file names with a language code. Then, it is possible to use heuristics to deduce translation file blocks from URL lists. For example, using a Portuguese web URL list (18 217 452 links) this method inferred about 50 000 blocks.

The third method is based on tests done with a set of files downloaded to a local directory. These tests include language identification, size comparison, non-text content comparison and some more tests.

Finally, we can detect pairs of files which points at each other: a page in Portuguese with a link for the English page and vice-versa.

1.2 Aligning with NATools

NATools is a set of tools to work with parallel corpora. It includes:

- a vanilla sentence aligner (Gale and Church, 1991; Danielsson and Ridings, 1997);
- a word aligner (Hiemstra, August 1996);
- corpora pre-processors (see section 2.5);
- integrated navigation system over translation dictionaires and parallel corpora;
- translation evaluation scripts;
- a word sequence aligner function;
- miscellaneous Perl scripts;

The tools are written in C and Perl. Aligners are written in C for speed and efficiency, and web CGI's and scripts are written in Perl for flexibility.

Section 2 presents the word aligner tool. Subsection 2.1 discusses the internal architecture for the aligner, showing its data flow while subsection 2.2 explains the structure of translated dictionaries created by the aligner. The next two sections, 2.3 and 2.4 present respectively times of the alignment process and an analysis of the resulting dictionaries. Subsection 2.5 shows an interesting (although naive) method to detect multi-word translations using only the word aligner. Finally, subsection 2.6 explains how translation dictionaries can be added together to produce better (and bigger) dictionaries.

Section 3 shows how the translation dictionaries created and the corpora can be useful.

2 NATools Word Aligner Tool

The aligner tool is based on Hiemstra's Twente aligner(Hiemstra, 1998). This tool uses statistical methods to create bilingual dictionaries.

2.1 Internal Architecture

NATools works counting co-occurrences of words on the same sentence and constructing a sparse matrix where co-occurrences are marked. Figure 2 shows the alignment process, and its explanation follows.

Given two corpus files (CorpusA and CorpusB) we process them with a filter to prepare the text.

This process tokenizes and normalizes the text. This stage can be used to remove some very common words (to reduce used memory and increase other word probabilities) or to process some multi-word terms. For example,

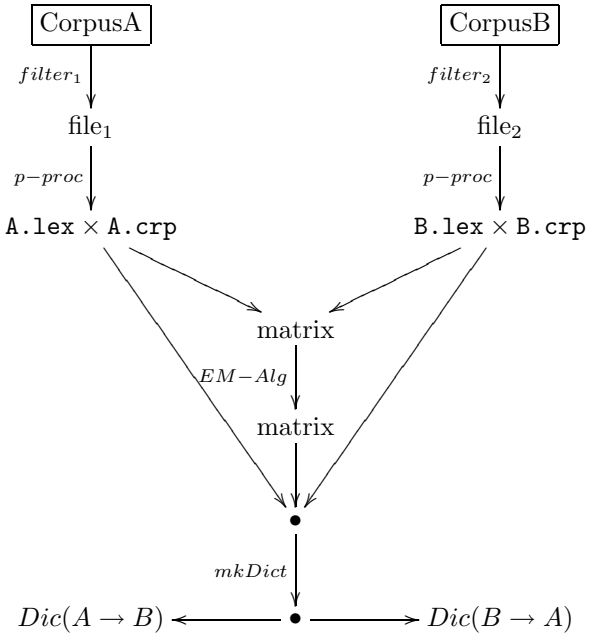

Figure 2: NATools data flow structure

we know some language constructs are multiterm as infinitive verbs ("*to see*", "*to read*") or genitive ("*dog's* mouth") in English. In section 2.5 we show how this method can be used for better results.

The following step is to process the prepared files ($file_1$ and $file_2$) creating auxiliary files for better efficiency when processing the corpora. Each text file will generate a lexicon file (`A.lex` and `B.lex`) with the corpus words and their correspondent integer identifier, and a corpus file (`A.crp` and `B.crp`) where text words are replaced by their identifiers;

Given the two corpus files (`A.crp` and `B.crp`) a sparse matrix is created. The matrix row number is the number of different words on the source corpora. The matrix column number is the number of different words on the target corpora.

Follows an iterative algorithm named Expectation Maximization algorithm (EM-algorithm) which tries to remove noise from the matrix enhancing the coordinates relative to the correct translation. This algorithm is explained with some detail on Djoerd Hiemstra master thesis(Hiemstra, August 1996).

Finally, the matrix is processed to create the two dictionaries, extracting points with higher value in the matrix. These dictionary structure is described bellow.

2.2 Created Dictionaries

Because the alignment process takes a lot of disk space to create auxiliary files, to deliver a package with all files is not viable. The proposed solution is to use a compiled dictionary which can be used alone for many purposes.

This dictionary is constructed over a Berkeley DB file which makes word access times reasonable, and can be read on almost any system. The DB file has the following definition:

$$w_\alpha \mapsto (N_{\text{occ}} \times w_\beta \mapsto P(\mathcal{T}(w_\alpha) = w_\beta))$$

where $w_\alpha \in \mathcal{C}_\alpha \wedge w_\beta \in \mathcal{C}_\beta$, $\mathcal{C}_\alpha$ is the source corpus and $\mathcal{C}_\beta$ the target corpus. So, the DB file maps to each word from the source corpus (w_α) a pair of information: the number of occurrences of that word on the source corpus (N_{occ}) and another map, from possible translations on the target corpus (w_β) to the probability of it being a correct translation – $P(\mathcal{T}(w_\alpha) = w_\beta)$.

The alignment process creates two of these dictionaries which can be used independently.

2.3 Measures

Although the general idea of Twente aligner was working, the system was very slow. One of the TerminUM design goals is to have tools strong enough to deal with real examples. This lead to code profiling: data structures were redesigned and some code re-implemented gaining a lot of efficiency as shown on table 1.

	Twente	**NATools**
Corpus analysis	180 sec	4 sec
Matrix initial.	390 sec	21 sec
Iterative method	2128 sec	270 sec

Table 1: Efficiency comparison using a parallel bible (Portuguese – English) on a Pentium IV 1.7 GHz,256 MB of RAM, running Linux

As a term of comparison we show table (results calculated on a Pentium IV 1.4 Ghz, 512 Mb of RAM, running Linux) with five different sized corpus:

- **Tom Sawyer** (TS) – the classic from Mark Twain: Portuguese – English;
- **Harry Potter, and the Sorcerer Stone** (HP) – first book from the Harry Potter series: Portuguese – English;

- **Anacom** (IPC)– an automatic generated corpus from the parallel web-site from the Portuguese communications authority: `http://www.ipc.pt`. The corpus was not manually reviewed and as such, contains noise. This explains the times lower than the other smaller corpus; Portuguese – English;
- **Bible** (Bib)– a parallel bible, where none of them is a direct translation of the other (they are parallel translations from the Greek and/or Hebraic): Portuguese – English;
- **EuroParl** (EP) – this English-French parallel corpus of EU documents was compiled and aligned by Andrius Utka at the Centre for Corpus Linguistic, at the University of Birmingham. On this test use used only half corpus: 139 825 sentences, 3 295 215 English words versus 3 705 784 French words;

	TS	**HP**	**IPC**	**Bib**	**EP**
k words	77	94	118	805	3 500
Analysis (sec)	0.5	1	1	5	67
Occurr. (sec)	6	8	4	57	893
EM-Alg. (sec)	42	73	44	468	5 523

Table 2: Time comparison for five different corpora

The EuroParl corpus is too big for aligning on our machines (the co-occurrence matrix takes more than 500MB of memory). This obliged us to split it on two halves. That lead to the idea of aligning portions of the corpus and add generated dictionaries. On subsection 2.6 we present the general idea for this tool, already used to sum-up both parts of this corpus.

2.4 Analysis of Results

The presented method builds two dictionaries mapping words from one language to a set of words in the other language. This set includes for each translation its probability of being a correct translation. Table 3 presents an excerpt from the dictionaries generated with the bible corpora.

It is important to recall that the mail purporse of the dictionaryis not to be a correct translation dictionary but a semantic web to help translators and other trained personnel to choose translations.

Deus		**God**	
God	0.86	Deus	1.00
(null)	0.04		
God's	0.03		
He	0.01		
Yahweh	0.01		
...	...		

gosta		**loves**	
loves	0.43	ama	0.67
detests	0.29	gosta	0.08
likes	0.29	amas	0.05
		estima	0.03
		...	...

Table 3: Resulting dictionaries from the word alignment for the Bible

Translations for the word "Deus" show a set of specific issues in word alignment. The first case is the normal one — the correct translation. The second line, "(null)" reflects a difference between English and Portuguese language types: in Portuguese we can omit the subject. The third line shows a case which can be solved pre-processing the corpus. As discussed in previous section, the English genitive constructions are, in fact, two words. If we split this term in two words the translation "God" would get higher probability. The last four entries are not so interesting.

Regarding the "gosta" translations we can see three entries. Two of them are normal translations: "loves" and "likes". The "detests" translations (which is the opposite of the correct word) appears because in Portuguese we say "não gosta" and the word "não" has a strong connection with "not" (disappearing from the correlation with "gosta"). Although this is not a correct translation it can be very useful.

For this same Bible (which is not properly parallel corpora) we extracted pairs of words (w_α, w_β) such that possible translations for each of one them includes the other with a probability above 70%. Analyzing the first 200 pairs we found about 160 correct translations (about 80% of correctness).

2.5 Detecting Multi-word Translations

It is known that some multi-word terms have a specific translation. For example, the "Natural Language" term is translated as "Lin-

guagem Natural" or "Computer Graphics" translated as "Computação Gráfica". While the first is translated almost "word by word", the second is a term which gained a specific meaning but not translatable "word by word".If we can extract these relations, we can build more useful dictionaries.

With this in mind we built a simple tool to construct pairs of words based on a corpus file. This is done in the filter phase: for each sentence, we put a token in the beginning of it and another at the end. Then, we join words, two at a time. For example:

```
I myself came weak ,
fearful and trembling ;
```

would result into

```
BEGIN_I I_myself myself_came
came_weak weak_, ,_fearful
fearful_and and_trembling
trembling_; ;_END
```

Processing two corpora in this form with the aligner would result on a dictionary translating pairs of words into pairs of words. Table 4 shows some examples of the output dictionary for the bible.

Jesus Cristo		**Christ Jesus**	
Christ Jesus	0.67	Jesus Cristo	0.94
Jesus Christ	0.26	(null)	0.04
(null)	0.03	Cristo ,	0.01
Messiah ,	0.01		
Christ who	0.01		
the Messiah	0.01		

um pouco		**a little**	
a little	0.68	um pouco	0.54
(null)	0.19	(null)	0.27
a while	0.03	Pouco depois	0.06
me a	0.03	e ,	0.03
your company	0.02	uma criança	0.03
BEGIN Then	0.01	BEGIN Daqui	0.02

Table 4: Resulting dictionaries from the word pair alignment

Looking to the result there are some entries (as "Christ Jesus") where the translation is correct with an high probability. On some other cases the translation is incorrect. In the table we can see "a little" to be correctly translated to "um pouco". Other translations, as "Pouco depois" can be explained with the "after a little time", or "uma criança" explained by the expression "a little child".

We should notice this test was done because its implementation was simple and could give some interesting results. A natural problem is that we only can find pairs correspondences, although in real examples pairs of words can translate to only one or more than two words.

Bigger tuples were tried but process time increase logarithmically with the number of words we put together in the tuple; matrix alignment became too huge to be usable; multi-words were almost not found.

2.6 Dictionary Addition

If one aligns various corpora where the source and target languages are the same, there is the possibility to add them, creating a bigger and, better dictionary. This is done adding the occurrence for each word and, for each word of the possible translations' list use the following formula:

$$\frac{\mathcal{P}_1(w_\alpha, w_\beta) \times \frac{\#_1(w_\alpha)}{\mathcal{S}_1} + \mathcal{P}_2(w_\alpha, w_\beta) \times \frac{\#_2(w_\alpha)}{\mathcal{S}_2}}{\frac{\#_1(w_\alpha)}{\mathcal{S}_1} + \frac{\#_2(w_\alpha)}{\mathcal{S}_2}}$$

where:

- $\mathcal{P}_n(w_\alpha, w_\beta)$ is the probability reported on dictionary n for the plausibility of w_β being a translation of w_α;
- $\#_n(w_\alpha)$ is the occurrence counter for word w_α on dictionary n;
- $\mathcal{S}_n$ is the number of words on the dictionary n;

This formula has the following advantages:

- uses only information contained on the dictionary files;
- the translation probability is calculated using the number of occurrences and the total size of the corpus to give different weights for the two different corpora.

3 Applications

As presented on subsection 2.2 the NATools generated dictionaries are very different from the normal translation dictionaries. This means that their use is different too.

This section shows some applications for these dictionaries, and aligned corpora: *(a)* integrated tool to search on parallel corpora, linked to a translation dictionary navigation system; *(b)* automatic translation evaluation; *(c)* a statistical translation tool;

3.1 Parallel Corpora search

This is a common tool found for parallel corpora. It lets you search for words occurrence on the corpora and see corresponding translations. It uses the processed corpora files and has the advantage of being easy to integrate with the other NATools scripts. Figure 3 shows a sample.

3.2 Browsing the Dictionary

The resulting dictionary is created on a text file with Perl syntax for data structures, which can be included directly on any script. The use of Perl syntax, and to have the dictionaries on different files makes it hard to browse and study the results.

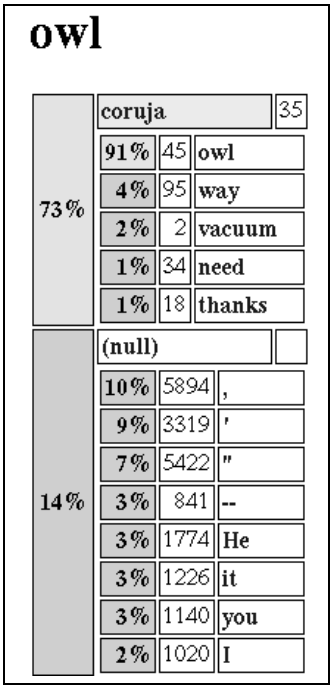

Figure 4: Dictionary navigation for the Harry Potter Corpora

This lead to the development of a simple CGI script to make possible web navigation over it. The navigation system give colors and sort the translations by probability. This and the shading of words when its translation translations includes the original word ($dic_1(dic_2(a)) = a$) results in an easy to read report. Figure 4 shows two screen-shots of this navigation system.

This navigation system is integrated with the previous one (for search on parallel corpora) being a common environment for parallel corpora and translation dictionary queries. It is possible to jump directly from this navigation system to a query for occurrences of the word and respective translation on the corpus.

3.3 Automatic Evaluation of Translations

One of the main decisions in TerminUM project is that each resource should be reused in other tasks in order to validate or improve them. The resulting dictionaries are being used to rank translations. In fact, the simple act of checking for each word w_α from sentence s_α it one of its probable translations $w_{i,\beta}$ is on the translated sentence s_β can give very interesting results.

To evaluate the translations we compute a value based on the words which have one of the possible translations on the target sentence. For each word in the source language we check if one of its translation is on the target sentence. Finally, we compute the weighted mean of these values (based on the number of words in the sentence). This process is done in both ways, resulting the mean of both values.

Table 5 shows an use example of this tool. We are using this tool in various tasks:

- validation of candidate pairs on web mining — after the download of files we use several tests before accepting them as a parallel text.
- filters over TMX files — we can validate each TMX translation unit and remove low values, or sort them by the translation rate;
- parallel corpora creation sorted by translation rate — CQP(König, 1999) tool takes a parallel corpora and maintains its order. This means that when looking to the results they are in the order they occur in the corpus. If we sort the corpora by translation rate we will probably get best translations first when consulting it.

3.4 Statistical Translation by Word Sequence Alignment

Created dictionaries, when associated to the corpora files can be used to translate sentences aligning them with previously word and sentence aligned corpora.

This alignment is done to a sequence of words, using a statistical approach.

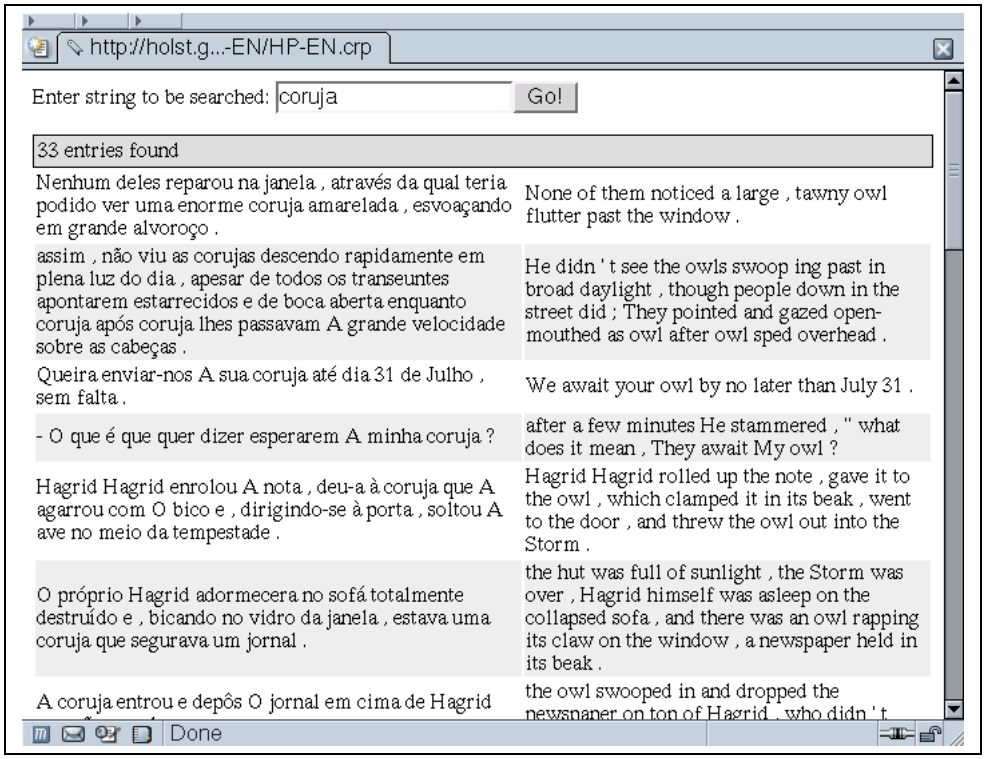

Figure 3: Parallel corpora search for the Harry Potter Corpora

Português	English	P(x)
Paulo, Apóstolo de Jesus Cristo por vontade e chamamento de Deus, e o irmão Sóstenes	From Paul, called to be an apostle of Christ Jesus by the will of God, and from Sosthenes, our brother	0.88
Pois em Jesus é que recebestes todas as riquezas, tanto da palavra como do conhecimento	For you have been fully enriched in him with words as well as with knowledge	0.18

Table 5: Translation evaluation example

To explain it, let consider a sentence and word aligned parallel corpus from $\mathcal{L}_\alpha$ to $\mathcal{L}_\beta$ and a sequence of words we want to align, $w^\star$ from $\mathcal{L}_\alpha$.

This sequence $w^\star$ is searched on the source corpus for occurrences. For each sentence s_α where $w^\star$ occurs, we find the respective translation sentence of the aligned corpus (s_β). If the alignment is correct, then s_β contains $\mathcal{T}(w^\star)$.

On this sentence, we use a sliding window algorithm, comparing the translation probability between each window and the original sequence $w^\star$ using the translation evaluation algorithm.

As this process is done to a set of samples, we can statistically calculate the better returned alignment (translation).

This alignment can be used as a translation tool as figure 5 shows — an interaction with a translation shell using the EuroParl corpus.

```
==> difficult situation
Using 6 occurrences (0.732864 seconds)
  situation difficile       - 0.8025
  situation très difficile  - 0.8025
  situation aussi difficile - 0.8025

==> sentenced to death
Using 1 occurrences (0.214145 seconds)
  condamné à mort - 0.443333333333333

==> final version
Using 7 occurrences (0.843922 seconds)
  version définitive - 0.5075
  définitive         - 0.09
  définitif          - 0.0875
```

Figure 5: Statistical translation example

By default, the script searches all the corpus for occurrences; this can lead to much time of search. To solve this, the corpus is previously ranked (using the automatic sentence evaluation method) and only a n samples are searched on the corpus, searching by translation quality.

4 Conclusions and Future Work

Hiemstra's work was very important because provided a framework to work with. Being a GPL(Free Software Foundation, Inc, 1991) program made it possible to study, reuse and recode some tools. The speed and memory improvements are giving us the chance to try to solve more complex problems and to test some other hypothesis we were unable to test before. As an example of the new experiments we have shown the bi-grams alignment.

All these tools can be found on Project Natura homepage at `http://natura.di.uminho.pt`. At the moment of writting the tools to mine the web for parallel corpora is not available, but word aligner and applications scripts can be downloaded freely.

Future work include:

- the use of a morphological analyzer to normalize dictionary entries and produce better relationships between terms. Some experiments were already done, normalizing Portuguese verbs to infinitive, which creates better relationships between them;
- comparison and/or integration with Kvec(Fung and Church, 1994) for word alignment and translation dictionary extraction;
- test *easy-align* with a list of word pairs created using the translation dictionaries to compare results with the pure alignment method.
- use a multi-term detection algorithm (statistical, for example), join them and re-align to find better multi-term alignment;

References

Almeida, José João, Alberto Manuel Simões, and José Alves Castro. 2002. Grabbing parallel corpora from the web. Number 29, pages 13–20. Sociedade Española para el Procesamiento del Lenguaje Natural, Sep.

Danielsson, Pernilla and Daniel Ridings. 1997. Practical presentation of a "vanilla" aligner. In *TELRI Workshop in alignment and exploitation of texts*, February.

Free Software Foundation, Inc. 1991. GNU General Public License, June.

Fung, Pascale and Kenneth Church. 1994. Kvec: A new approach for aligning parallel texts. In *COLING 94*, pages 1096–1102, Kyoto, Japan.

Gale, William A. and Kenneth Ward Church. 1991. A program for aligning sentences in bilingual corpora. In *Meeting of the Association for Computational Linguistics*, pages 177–184.

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.

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

König, Oliver Christ & Bruno M. Schulze & Anja Hofmann & Esther. 1999. *The IMS Corpus Workbench: Corpus Query Processor (CQP V2.2): User's Manual*. Institute for Natural Language Processing, University of Stutgart.

OSCAR. 2003. Open Standards for Container/Content Allowing Re-use — tmx home page. `http://www.lisa.org/tmx/`.

Resnik, Philip. 1998. Parallel strands: A preliminary investigation into mining the web for bilingual text. In *D. Farwell, L. Gerber, and E. Hovy (eds.), Machine Translation and the Information Soup (AMTA-98)*. Lecture Notes in Artificial Intelligence 1529, Springer.

Resnik, Philip. 1999. Mining the web for bilingual text. In *37th Annual Meeting of the ACL'99*. College Park, Maryland.

Savourel, Yves. 1997. Tmx 1.4a specification. Technical report, Localisation Industry Standards Association.