Using WebCorp in the classroom for building specialized dictionaries

Natalie Kübler

University Paris 7

Abstract

In this paper, we present an experiment that was led to use finite corpora and WebCorp in the classroom with a peadagogic objective that was different from language teaching. WebCorp use and corpus use were embedded in the wider frame of teaching students how to use machine translation, by building a customised dictionary using available tools and resources. The issue of using finite corpora and the Web as a corpus was raised in that frame, and will be discussed. Although there is no simple and definite answer, this issue led students to put the Web into question, as a source or information, and to better understand the issues at stakes in corpus building and corpus use.

Introduction

In this paper, we present an experiment that was led using finite corpora and WebCorp in the classroom with objectives that were different from mere language teaching. Corpus-based, or corpus-driven teaching, as Johns (1988) introduced it, can be adapted to using the Web as a corpus; in this case, WebCorp can be a useful tool for language teachers and students. Our purpose was however slightly different. Although WebCorp was tested in a pedagogic situation, its use was embedded in the wider frame of teaching students how to extract lexical and syntactical information to build customised dictionaries for machine translation (MT) in languages for specific purposes (LSPs). In the light of this specific context, we shall tackle the issue of finite corpus use as opposed (or not) to WebCorp use.

The first part of this paper presents the pedagogic and scientific context of the experiment. Some details must be given about the project in which the experiment took place, since it has an impact on the type of results that were expected from the WebCorp search.

In the second part, the resources and tools that were used are described.

In the third part, samples of the results obtained with WebCorp and with the finite corpora will be presented and explained. We will show how WebCorp can be used to complement and update search for linguistic information infinite corpora. This part will also discuss the benefits of using WebCorp parallel to querying finite corpora.

The conclusion will deal with future prospects and enhancement requirements for WebCorp.

Experiment Context

The experiment took place in a postgraduate syllabus[1] called "Language Industry and Specialised Translation". This syllabus is oriented towards computer-mediated translation. Students have thus courses in four specific areas, mainly

- translation: theory and practice;
- linguistics : syntax, corpus linguistics, terminology;
- cultural studies;
- technology: database management systems, HTML, XML, translation memory, localisation tools, and machine translation.

This translation training is semi-professional since students spend every other week as interns with a private company.

WebCorp was used in an introductory course to corpus linguistics and its application to translation and terminology. As the best way of training students is to place them in real-life situations, they had to realize translation projects in the subject area of computer science. Part of the projects consisted in building customised dictionaries for machine translation. Students were first shown how to extract lexical and syntactic information in the source and target language, from comparable and parallel corpora. They then experienced extracting linguistic information from the Web using WebCorp. The two approaches were applied to dictionary building.

Pedagogical Objectives

The objectives of this project consisted not only in teaching the students the various skills which will be described below, but also in considering the limits of finite corpus use versus "Web as a corpus" use. This approach is very profitable to young people who are computer-literate, and to whom the Web is close to the unique reference for truth. Comparison helps them find the advantages and disadvantages of the two approaches; it is also aimed at showing them that information extracted from the Web must be carefully examined, and not be taken for granted. This also raised the issues at stake in corpus-building as opposed to using texts collected without specific criteria, or using the Web

Below are listed the competences students should have acquired at the end of the course; they should be able to:

- use a machine translation (MT) system and add appropriate bilingual dictionaries to improve translation results;
- use available term extraction tools, which do not require particular computing skills;
- use available resources, such as Web-based bilingual glossaries, self-made or Web-based finite corpora and the Web as a corpus;
- proofread translation results to produce a professional translation;

[1] The French DESS (Diplôme d'Etudes Scientifiques Spécialisées) which is equivalent to the second year of a "vocational" M.A.

- analyse the system's translation "errors" from a linguistic point of view, in order to grasp which very delicate linguistic issues are at stake in MT. This shall show students how important the human factor is, whatever tools and resources are available for each part and step of the translation process.

The whole range of competences was included in the translation project that will be described below. The whole workflow of translating documents with customised machine translation in which corpus use is predominant is fully described in Kübler (2002).

Project Description

The projects in which WebCorp was used and tested consists in translating texts in the computer science area, using a customisable machine translation system. Some texts to be translated from English into French were dictionary definitions, extracted from a Web-based computing dictionary[2]; the other type of texts were some of the Linux HOWTOs that have not been translated yet. The Linux HOWTOs are the "user manuals" of the Linux operating system[3].; they have been translated into several languages by the various Linux communities. The French Linux community is quite active and has translated most HOWTOs. However, as new HOWTOs, or updates of previous ones, are regularly released, there are still some documents that remain to be translated yet. Our students had thus to translate some of the most recent HOWTOs.

The machine translation system that was used was Systran, and more precisely Systranet which is Systran's customisable on-line translation system. It allows users to create their own (bilingual or multilingual) term bases to improve translation results; this feature can give quite good results in specialized translation. Students had to create their own customised dictionaries, in order to test them with Systranet.

To create term bases (or customized dictionaries) from scratch, the first step consisted in automatically extracting term candidates from the English text to be translated and then in finding their French equivalents. The first dictionary would then be used to translate the text.

Systranet offers the possibility of aligning the source and target text, and, in the aligned target text, highlights unknown terms in red and the user's dictionary terms in green. These features make it thus possible for the user to add to the dictionary all the words that are not recognized by Systran's home dictionaries. The second step is more demanding in terms of linguistic work: students compare source and target texts to complete and modify the dictionary until no more dictionary change can improve the translation result. When the dictionary is saturated, i.e. no more change can be made to improve the translation result, the final translation of the text is launched; the result will then be proofread and post-edited to correct the translation errors that could not be solved by modifying the

[2] FOLDOC: Free On-Line Dictionary of Computing.

[3] Linux is a Unix type operating system that is freely available to the community.

dictionary.
Finite corpora and the Web as a corpus are deeply involved in the process of building and correcting dictionaries, and of proofreading the final translation result. After extracting term candidates from the source texts, students must decide which candidates are actual terms. Corpus query must then be applied to answer this question. Parallel corpora are then necessary to help find the French equivalents for the terms. Corpus use is not only essential to find terms and their equivalents, it is also often the only possible solution to find syntactic information for the terms, especially for verbs and adjectives; verbs and adjectives indeed are not always considered as terms, therefore little linguistic information can be found.
Finite corpora are not the only resources that are essential to creating customised dictionaries; it will be shown later, how the Web as a corpus can complete and update the information extracted from finite corpora.

Tools and Resources

This section deals with the description of the tools and resources that were used to fulfil the assignments in the project. The two most important resources for the issues at stake in this paper are WebCorp and the finite corpora that were used.

WebCorp

WebCorp is project that was set up at the Research and Development Unit for English Studies at the University of Liverpool. Its objectives were to investigate the usability of the Web as linguistic resource. The project also had to identify and address problems of retrieval and analysis. It allows the user to type in a request for linguistic information that is processed and fed to the selected Web search engines. The search engine returns a list of URLs that WebCorp accesses directly; it then returns concordances, or collocates for the query. We will show below how it can be used to retrieve useful linguistic information to create bilingual term bases in LSPs. A complete description of WebCorp has been given by Renouf (forthcoming) and Kehoe and Renouf (forthcoming).

Corpora

The finite corpora that were available for the students were first developed at the Laboratoire de Linguistique Informatique at the University of Paris 13. There have been augmented and enhanced at the University Denis Diderot Paris 7 for several years. Those corpora, parallel and comparable, and are accessible via a Web-based interface[4], on which a concordancer allows visitors to use perl-like regular expressions, as described in Foucou & Kübler (2000). Below are the corpora that were used by the students:

[4] http://wall.jussieu.fr

a) The parallel English-French HOWTO corpus, that has been used for several years et Paris 7. It is made of the Linux HOWTOs ("user manual" files of the Linux operating system), which have originally been written in English. The HOWTOs have been translated into several languages, among those into French. The source language and target language texts were aligned at the section level. The size of the parallel corpus is approximately 500'000 words each. It is possible ask for concordances and then have an aligned view of the section in which the term, or the expression, was found. Concordances with regular expressions are very useful to extract refined linguistic information about terms. Then, being able to look at the equivalent section in French leads to finding the French equivalents of the term or expression.

b) Smaller comparable corpora in English and French on subdomains of computing (less than 100'000 words), such as artificial intelligence, peripherals, computer games, digital cameras, etc. were also made available to the students. This led us to develop a methodology for querying comparable corpora to extract French equivalents for an English term.

c) Our students used an experimental version of WebCorp that gives access to additional features, such as regular expressions, and domain filtering. This was particularly useful as the students were working in a specific subject area, namely computer science.

Tool: Machine translation

Apart from the university developed Web-based interface for corpus query, and from WebCorp, the other tools can be found on the market, as is the case for Systranet[5] and Terminology Extractor[6].
Systranet is an on-line machine translation system, developed by Systran. It gives access to Systran's over 35 language pairs, and allows users to translate either a text file, or a formatted file, or a Web page. Users can create their own customised dictionaries and compile these into the system to help them translate specialised texts. Users can work in a network of translators, each member of a group having access to the other members' dictionaries. The interface we used was adapted to specific pedagogical needs, allowing the teacher to create the groups, and to have access to all the students' dictionaries, as well as partially to the logs of the sessions.
The most interesting feature for our project, apart from the translation engine as such, was the possibility for the user to create and compile customised dictionaries.
Dictionaries contain more than just a correspondence between a source word, namely in English here, and a target word, namely in French, since users can

[5] http://www.systranet.com
[6] http://www.chamblon.com

enter what is called 'advanced' linguistic information in these. The allowed information can be divided into several levels:

part-of-speech information: basic part-of-speech information can be attached to the entries, such as verb, noun, proper noun, adjective, and "sentence", which deals with adverbs, adverbial phrases, or whole idioms, such as *your mileage may vary*.

syntactic information, such as the governed prepositions for nouns, verbs, and adjectives, or direct objects for verbs. A verb which governs a preposition is shown in example (1).

(1) access (verb)(noprep)=accéder (verb)(prep:à)

semantic information, such as the conceptual class of the possible direct object of a verb, as shown in example (2). In this example, the coding for the verb *run* indicates that the direct object must belong to the semantic class [OS], which means all terms sorted under the 'operating system' class. Below the verb, the noun *Unix* is marked as belonging to the [OS] class. This means *Unix* can be the direct object of *run*.

(2) to run (verb)(context:OS)
Unix (noun) (SEMCAT:OS)

morphological information, such as the plural form of a noun in any language, the gender of a noun in French, or forcing the number in the target or source language. Example (3)shows how the gender of *cache* can be forced to masculine. In general French, the noun *cache* ('hiding place') is feminine, whereas in computer science French, it is masculine, and means 'cache'.

(3) cache(noun) = cache (noun) (masculine)

The term *URL* builds a plural in *–s* in English, i.e. *URLs*, whereas in French, it is invariable; this type of information can be coded in the dictionary, as is shown in example (4).

(4) URL (noun) (plural:URLs) = URL (noun) (plural:URL)

translational information, such as "DNT", which means that the string must not be translated, must stay as it is in the translation process. This feature is quite useful in computer science, as there are command names for example that are never translated, such as the Unix command *cd*, or *mkd*.

Figure 1 shows a dictionary sample, in which various types of coding are presented.

"AT&T" (company name)
auto-dial (noun)=numérotation automatique (noun)
automatic number identification (noun)=identification de l'appelant (noun)
based (adjective)(noprep)=architecturé (adjective)(prep:autour)
basic language constructs (noun) (plural)=base de construction du langage (noun) (singular)
to log in (verb)=se loger (verb)
to introduce (verb) (context:extensions)=introduire
to carry (verb)(context:digital data)=transmettre (verb)

Figure 1: dictionary sample

Tool: Term extraction

To extract term candidates from the source texts, a very simple and user-friendly tool was applied, Terminology Extractor. This tool works for English and French, and gives several types of results. First, it extracts all the words that are recognised by its dictionaries, plus all the non-words, i.e. that are not in the dictionaries. The non-words feature is interesting, as it usually gives a list of very specialised words, which are not in general dictionaries. Then it extracts in a window of two to ten words, all the sequences that have been repeated at least once in the text, i.e. that appear at least twice. This feature allowed the students to have a list of term candidates among which they chose the actual terms with the help of the various corpora and WebCorp.
A sample of the term extraction results is given in Figure 2 and 3. Figure 2 contains the result of the non-words extraction, and Figure 3 the result of the 'collocations' extraction. It shows that an important linguistic job must be done on the results to obtain an actual list of terms (single and compound).

Debian	Netscape	accellerate
Permedia	Dennis	XFCE
RedHat	Dialogs	Corel
RgbPath		FAQs
ServerFlags	Howto	Microdoft
ServerLayour	README	Linux
XkbLayout	XkbModel	RealAudio
Solaris		ISA
UI	KDE	GUI
USB	LeftOf	IRQs
WindowMaker	ModulePath	NFS

Figure 2: Result of the non-words extraction from a HOWTO document. Apart from 'Dennis' and 'accellerate', all the words are terms or product names in the computer science area.

Internet Gateway 3	{ Looking look } at the Network 3
IP aliasing 3	**name server** 4
ISA { card cards } 3	**Network { Device devices** } 4
latest version 3	**Linux computer** 3
DHCP Server 15	IP **{ addresses address** } 16
Linux gateway 3	**Linux box** 16
modules file 3	card on the Linux box 4
scripts / ifcfg 3	**DNS { Server servers** } 17
server will start 3	**interface configuration file** 3
{ **Network networking } { Card Cards** } 12	

Figure 3: Results of a collocations extraction from a HOWTO document. The wordsin boldface are actual terms.

Other information sources

Finite corpora and the Web as a corpus were the main resources used in the project. There were secondary sources, such as on-line glossaries, or on-line term bases. Those were presented to the students to help them understand why data-driven information is essential to this type of work, and why dictionaries and glossaries are not satisfying in that case. Figure (5) shows the type of information that can be accessed in a Web-based bilingual term base. The search for the translation of the English word *buffer* brought the translation *mémoire*-tampon, and three synonyms and translations for those, but no syntactic and phraseological information. There were no compounds of the word 'buffer', although it is very productive in computer science English.

ENGLISH	**FRENCH**
buffer	mémoire tampon n. f.
Syn.	**Syn**.
buffer storage	tampon n. m.
buffer memory	mémoire intermédiaire n. f
intermediate memory	zone tampon n. f.

Figure 5: the term *buffer* and its French translations in 'Le Grand Dictionnaire Terminologique'.

Finite Corpora and WebCorp use

Taking our experiment in the classroom into account, we want to show how the use of finite corpora and WebCorp are neither contradictory, nor incompatible. Available finite corpora, such as the HOWTO corpus, and the smaller ones in subdomains of computing can give the user a lot of information. But, as computing is a very quickly changing domain, new terms are coined all the time, which means that available corpora can become insufficient, or slightly obsolete, even though they can be regularly updated. In the subject area of computer science, most neologisms can be found on the Web. So being able to query the Web as a non finite corpus seems appropriate to complete missing information.
Taking the above-mentioned example of *buffer*, we will describe and discuss this.

***Buffer* in the HOWTOs**

As shown in Figure 5, the term *buffer* is translated into *mémoire tampon* in French. However, 'Le Grand Dictionnaire Terminologique' did not mention any compound for this term. Looking for *buffer* in the HOWTO corpus results in finding several multi-word units. Looking at the aligned section in French

allowed us to find French equivalents for those, as shown in Figure 6.

buffer cache (noun)	mémoire cache (noun)
buffer memory management (noun)	gestion de la mémoire tampon(noun)
buffer store (noun)	zone tampon (noun)
DRAM write buffer (noun)	buffer d'écriture DRAM (noun)
frame-buffer (noun)	tampon de trame (noun)

Figure 6: Multi-word units for *buffer* and their French equivalents.

The problem is that the HOWTOs translators have not always translated the whole text, or have modified sentences in such a way that some words just disappear. So some compounds can be found, but not all, and not always their French equivalents. This is were the limit of finite corpora stands. New terms that were created after the collection of the corpus, or translations that have been thoroughly modified cannot be found in a finite corpus. Term bases are generally not complete enough. In this case, the information must be looked for on the Web. As not only lexical information, but also phraseological and translational information is necessary, a tool allowing to extract concordances from the Web is mostly appropriate. The next sub-sections deal with examples of Web search, using WebCorp, and that show how the necessary information can be found.

WebCorp: searching for the French equivalents

As the Web is not an aligned corpus, heuristics must be applied to find the French equivalents for English words. One possibility consisted in searching for an English term on a French Web-site. In the current state of WebCorp, the only way of doing that was to look for URLs in the French domain, i.e. ending in .fr. In French, computer scientists often use the English term for a given concept. Some translators therefore use the English term and often give its French equivalent in parentheses at the beginning of the document and then no more. Others use the French term, but put the English word beside, in parentheses. So this permitted us to find translation and also more terms, as in Figure 7,which shows a concordance for *buffer* extracted with WebCorp. These concordances yields two multi-word units in English, and their equivalents in French, i.e. *buffer overflow* and *heap buffer overflow*.

me des débordements de **buffer** (tampon en français). Pour
com/advisories/bufero.html . Writing **buffer** overflow exploits – a tutorial for
de NOP . débordement de **buffer** dans le tas (heap buffer overflow)
(buffer overflow) . débordement de **buffer** sous windows (et oui ;-)) --[

Figure 7: Concordances for *buffer*.

Not all translation provide the reader with the English source term in parentheses. In the case of *dial-in line* for example, only part of the term is translated into

French, and no indication of the source term is given. Figure 8 shows an occurrence of *ligne de dial-in*, in which only part of the term is translated. However, other occurrences of *dial-in* in French text show that this is the correct way of using it in French.

Monter un **serveur PPP/POP dial-in** Par Hassan Ali AVERTISSEMENT : a
avec une des **lignes de dial-in PPP** et son adresse IP
assigner dynamiquement aux utilisateurs du **dial-in PPP**. Ceci, bien sûr
pouvez assignez vos **clients de dial-in :** # Secrets for authentication using PAP
Doe appelle l'aide de **l'adaptateur dial-in** de Windows 95 qui est

Figure 8: *dial-in* in French documents.

WebCorp: Searching for linguistic information: *To run*

As mentioned above, creating a customised dictionary for machine translation does not only require extracting lexical information from corpora, completed with using the Web as a corpus. Phraseological information is also essential to insert more advanced linguistic information in the dictionary. Furthermore, this type of information is important during the proofreading and post-editing process of the translation.

Terms of a domain have specific meanings that are usually unknown in general English. In computer science, the verb *to run* has a meaning that greatly differs from its meanings in general English. Not surprisingly, the French translation of the verb in computer science French, has nothing to do with its general meaning translation. When *to run* means 'to walk quickly', its French equivalent is *courir; to run* used in the computing world is translated by *tourner,* or *lancer*, or *exécuter*, which have nothing in common with *courir*.

To run in computer science can be followed by a direct object, then, either by the preposition *on,* or by the preposition *under*, usually depending on the type of argument that fill in the indirect object position. Example (5) shows samples of the syntactic structure:

(5) *You can run a program under an operating system*
You can run a program on a platform + OS

An argument that appears after the preposition *under* can also be used after *on*, but the opposite is quite rare. Building a customised dictionary means listing, as exhaustively as possible, the different verb arguments that can take the different positions in a sentence. Finite corpora can bring a quite exhaustive answer, which needs to be completed and updated by using the Web as a corpus. Figure 9 shows how the expression "**run * * on**", which uses two wildcards instead of words before the preposition *on*, can give significant results on the arguments that can fill out the syntactical positions. Those arguments could not be found in the HOWTO corpus, not in the smaller finite corpora

harm is done if you **run** cvs init **on** an already set-up repository.
containing all you need to **run** Tcl/Tk **on** a Macintosh. tcl8.0p2.tar
nd showed that it can **run** equally well **on** a Sharp or Alcatel telephone
you will be able to **run** PETSc ONLY **on** one processor. Also, you will
ith my favorites tools, and **run** the binary **on** a real ST. If the

Figure 9: Arguments for the verb *to run*

Another useful feature offered by WebCorp, is the collocates function; it gives the most frequent collocates of the sequence. Frequent collocates for the verb *to* run for example are *Debian, Alpha,* and *messages,* the first two being product names in computer science. *As* WebCorp is limited in the number of sites that can be opened, it is possible to filter the collocates out, in order to discard the URLs in which those collocates are. It can be done by using the 'exclude' feature (using the '-' sign, as in search engines). This allows WebCorp to extract concordances from other URLs, which then leads to more linguistic information.
The same operation can be applied to extract linguistic information for the French equivalent of the verb, i.e. *tourner*. As shown in Figure 10, the first pass is not always concluding, since there are occurrences that have nothing to do with computer science. The sequence 'tourn* * * sur' will find all the words beginning with *tourn*, followed by two words, followed by the preposition *sur* ('on').

First pass without filter apart from « .fr » and « computers »
état de conservation : Ce denier tournois est frappé
japonais. . n'a pas renoncé à tourner son film sur le
sterling bruce subspace sun open : tournoi de golf sur
d'éternité: quatre poules blanches tournant en rond sur une place de village et

Figure 10: Occurrences of *tourn* without any filtering out.

In the second pass, some filtering can be done to include keywords of computer science, such as *programme*, *système*, *Linux* and *machine,* and to exclude words, such as *film, napoleon,* or *poule* for example. In that case, the search result is much more consistent with the subject, as shown in figure 11.

fonctionner avec Windows, il peut **tourner** ou pas sur des cartes vidéo ou
de type Unix qui peut **tourner** entre autres sur PC. Il est installé par
des ordinateurs distants Pour faire **tourner** un programme sur une machine distante dont l'adresse
texte ASCII par un module **tournant** sous Windows (sur PC) et devrait bientôt

Figure 11: Occurrences of *tourn* using filters.

Discussion

These few examples show occurrences of terms or phraseological contexts that could not be found in the finite corpora on computer science. Studying terminology and phraseology for practical purposes raises issues that are different from describing the language as such. Describing languages for specific purposes means working in well-defined subject areas, which does not need huge corpora, as the study of general language (if there is such a thing as 'general language'). A few hundred thousand words, sometimes, less than a hundred thousand words are enough to describe the characteristics of a language for specific purposes. However, applying this type of description to practical purposes, such as creating a dictionary that will be compiled into a machine translation system, raises the issue of exhaustivity. Machine translation needs human input to give satisfying translation results. In this case, a small, specialised corpus is not enough. Moreover, the issue of up-to-date information is raised. WebCorp, as a tool enabling the user to follow daily updates, is ideal to complete and update the information extracted from specialised finite corpora.
However, using finite corpora presents some advantages on WebCorp, that will be difficult for a concordancer using the Web as a corpus to overcome. Finite corpora have the significant advantage of presenting a controlled and balanced information. The texts collected in a corpus have been selected among other candidates. Using the Web as a corpus implies that no control can be made on the content of the documents that are extracted. The huge quantity of documents is also the limit.

Conclusion

While, in our case, finite corpora would give the basics for the creation of customised dictionaries, WebCorp provided us with more complete and updated linguistic information. In the classroom situation, students were faced with those issues, i.e. finding information in finite corpora, discovering they would need more, and using WebCorp, instead of collecting a bigger corpus in the domain. Students learned how to use heuristics to find appropriate information using WebCorp; this also led them to note the advantages of WebCorp over classical search engines, namely, concordances, collocates, regular expressions, and the possibility of limiting and filtering the linguistic information.
WebCorp still needs some improvements, such as refining language identification, and domain filters. Linguistic information extracted with WebCorp would be more accurate if domain filters could be used, to restrict the search to one domain. Refined regular expressions would allow users to extract more accurate phraseaological information. As these improvements will be integrated into the next release of WebCorp, the next step is to test those and see if the

References

Foucou P.-Y. et Kübler N. 2000: 'A Web-based Environment for Teaching Technical English." in L. Burnard and T. McEnery (eds.) *Rethinking Language Pedagogy: papers from the third international conference on language and teaching.* Frankfurt am Main: Peter Lang GmbH. 65-73.

Johns, T (1988), 'Whence and wither classroom concordancing?', in T. Bongaerts, P. de Haan, S. Lobbe and H. Wekker (eds.) *Computer Applications in Language Learning*. Dordrecht: Foris. 9-27.

Johns, T. (1993), 'Data-driven learning: An update', *TELL & CALL* 3.

Kübler N. 2002: 'Creating a Term Base to Customize an MT System: Reusability of Resources and Tools from the Translator's Point of View'. In E. Yuste (ed.) *Proceedings of the Language Resources for Translation Work and Research. Workshop of the LREC Conference*. Las Palmas de Gran Canarias: ELRA. 44-48.

Kehoe, A. & A. Renouf (forthcoming) `Webcorp: Applying the Web to Linguistics and Linguistics to the Web'. In *Proceedings of the WWW 2002 Conference*, Honolulu, Hawaii, 7-11 May 2002.

Pearson, J. (1998) *Terms in Context*. Amsterdam: John Benjamins Publishing Company .

Renouf, A.J. (forthcoming). WebCorp: providing a renewable energy source for corpus linguistics, in S. Granger and S. Petch-Tyson, (eds) *Extending the scope of corpus-based research: new applications, new challenges*,. Amsterdam & Atlanta: Rodopi.