National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC · CNRC

# *TerminoWeb: A Software Environment for Term Study in Rich Contexts **

Barrière, C., Agbago, A.
August 2006

* published at the International Conference on Terminology, Standardisation and Technology Transfer (TSTT 2006). Beijing, China. August 25-26, 2006. NRC 48765.

Canada

# TerminoWeb: a software environment for term study in rich contexts

**Caroline Barrière**
National Research Council of Canada
Gatineau, Canada
*caroline.barriere@nrc-cnrc.gc.ca*

**Akakpo Agbago**
National Research Council of Canada
Gatineau, Canada
*akakpo.agbago@nrc-cnrc.gc.ca*

## Abstract

TerminoWeb is designed to provide a work environment for terminologists to help them in one of their multiple tasks, that of doing a thematic search toward the understanding and the definition of a set of terms for a new domain (theme). It is a time consuming task for the terminologist to read through several documents to get an understanding of a domain and apprehend its main concepts and their interrelations. The purpose of TerminoWeb is to provide functionalities for collecting knowledge-rich documents into a corpus, for discovering important terms, and for exploring knowledge-rich contexts around these terms in the corpus. Such work environment would allow terminologists to focus their attention more quickly on the important material to read.

## 1 Introduction

The purpose of the TerminoWeb platform is to offer an integrated environment for performing thematic searches. Such type of search usually has the objective of gathering information about a particular domain and providing a concise representation of that domain.  Such representation is often in the form of term records, in which each term is defined and possibly related to other terms.

We identify three tasks towards this goal: (1) finding knowledge-rich documents, (2) extracting important terms from such documents, and (3) extracting knowledge-rich contexts from the documents. Each task poses a certain number of challenges for its full automation, and TerminoWeb, in its current state, is designed as an interactive platform for achieving the tasks semi-automatically.

Overall TerminoWeb is a software environment to help in the process of knowledge gathering and structuring. Figure 1 gives a schematic view of TerminoWeb.

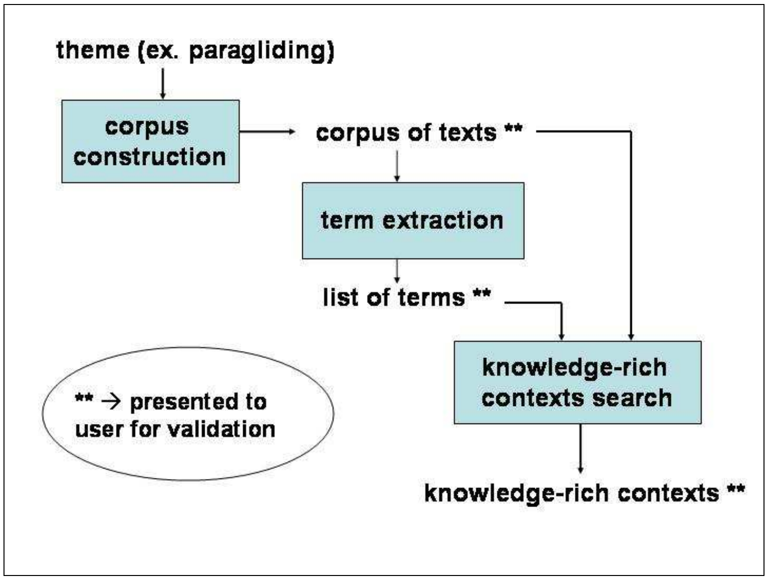

theme (ex. paragliding)
corpus construction
corpus of texts **
term extraction
list of terms **
knowledge-rich contexts search
knowledge-rich contexts **
** → presented to user for validation

Figure 1: Schematic view of TerminoWeb

The first task, corpus construction, is to find documents about a theme and filter the ones likely to contain more definitional knowledge than others. To understand and structure the knowledge for a new theme, we need to access and gather definitional knowledge, knowledge which conveys important information for the understanding of terms. This important information is often expressed in the form of semantic relations between terms. The semantic relations we are particularly interested in are synonymy, meronymy, hyperonymy and function. We want "good" documents identified to be part of our corpus to be rich in surface expressions of these semantic relations. Such surface expressions are also called knowledge patterns. In TerminoWeb, we develop a document filtering system to look at each document in the collection and rank it with respect to its potential for knowledge structuring.

The second task, term extraction, is to find important terms of the domain. This is a problem studied by many researchers in the field of computational terminology. For this task, we implement a state of the art approach within TerminoWeb platform. Without imposing a list of terms to the terminologist, TerminoWeb automatically extracts and suggests a list of single-word and multi-word candidates having a high potential of being validated as terms.

The third task, knowledge-rich contexts search, builds up on the results from the first two tasks. Being able to find documents containing a high number of knowledge patterns, and being able to extract candidate terms from these documents, we can further search for contexts that combine terms and knowledge patterns. Such contexts have been referred to in the literature as Knowledge-Rich Contexts (KRC). Presenting knowledge-rich contexts for a specific semantic relation, e.g. hyponymy, and a specific term, e.g. *diving*, would lead the terminologist to easily find a series of hyponyms for the term, e.g. *cave diving*, *technical diving*, *cavern diving*, *overhead environment diving*, *recreational diving*.

The remaining of this article presents each task. Section 2, the core of the present article, presents the corpus construction module in search for "good" documents. Section 3 briefly presents the term extraction approaches used. Section 4 presents the search capabilities of TerminoWeb to retrieve knowledge-rich contexts. Finally, section 5 concludes and presents future work.

## 2 Corpus construction - finding "good" documents

Suppose a new theme is brought forward by a client. The client is interested in the terminology of paragliding, or computer storage, or bank fraud. If the client provides lots of related documentation and texts, the terminologist would start by exploring these, but what if there were no documents? The terminologist could certainly start his/her work from nothing else than the seed words *paragliding* or *computer storage* or *bank fraud* to find documentation about the theme on the Web, but much of the retrieved information would not be very useful. The Web is an invaluable resource about all kinds of topics, but it is also very noisy. The problem is often not "whether the information is on the Web", but rather "how to find it quickly?". As shown in Figure 2, we state that valuable documents, from a terminological point of view, are documents with three quite specific attributes: specialized, knowledge-rich and containing a flowing text structure. These are neither exhaustive nor exclusive criteria and others (e.g. reliable source, date) are mentioned in [1, p.126ff]. In this research, we wish to focus on *measurable* criteria as we are aiming at automatic filtering. We explore each criterion hereafter.

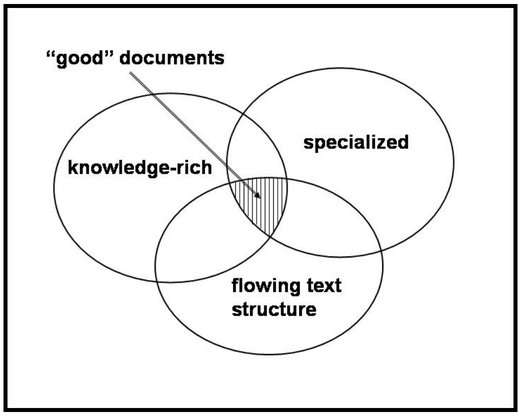


Figure 2: Intersection of 3 textual attributes

Knowledge-rich

Inspired from Ingrid Meyer's work [2], we define a knowledge-rich document as one with a high density of knowledge patterns (KPs). Each KP is associated to a semantic relation as explained in [3]. A few examples of semantic relations and KPs are presented in Table 1. In fact, TerminoWeb is flexible enough to allow users to decide on any semantic relation they wish to emphasize and to define any knowledge patterns they think are appropriate for such relations.

Table 1: Knowledge patterns

| SEMANTIC RELATION | KNOWLEDGE PATTERNS |
|---|---|
| Hyperonymy | such as, and other, or other, including, includes, is classified as, classified as, is a kind of, are kinds of, is a sort of, are sorts of |
| Meronymy | is a part of, are parts of, is made up of, makes up, comprises, has the following components, is a component of, is composed of, consists of, is a constituent of |
| Synonymy | known as, also known as, also called, is another word for |

Our previous work [4] explains and validates the hypothesis that we can differentiate a Google-made corpus (using the top ranked documents retrieved from Google search results) with terminologist-made corpus (corpus manually built by terminologists) using the KP density.

Specialized

For a document to be specialized, it should contain a sufficient density of terms from a domain. For example, a specialized document on scubadiving is assumed to have a high content of specialized terms about scubadiving. The problem, when starting on a new thematic search, is that the list of specialized terms is not an input of the task, but rather an output because we do not know the terms (except for maybe a handful) ahead of time. We have suggested in [4] an iterative approach to search for both specialized and knowledge-rich documents as a combined criterion. We will describe this idea in the following section on system design.

Flowing text structure

For the terminologist, the quality of a document relies on its content but also on its form. Our hypothesis is that a document having long sentences organized in paragraphs is likely to embed useful information for the comprehension of the theme as well as provide for easy reading. This is what we call a flowing text structure. See for example document B in Table 2 in opposition to the fragmented document A which has little lexical information.

Table 2: Examples of documents

| ***Document A: fragmented text extracted from an html document*** |
|---|
| U.S. PATENT DOCUMENTS 5,761,667 A 6/1998 Koeppen<br>5,392,390 A 2/1995 Crozier 5,778,346 A 7/1998 Frid-Nielsen et a!.<br>5,442,783 A 8/1995 Oswald et a!. 5,778,389 A 7/1998 Pruett et a!.<br>5,510,981 A 4/1996 Berger et a!. 5,813,009 A 9/1998 Johnson et a!.<br>5,519,606 A 5/1996 Frid-Nielsen et a!. 5,832,487 A 11/1998 Olds et a!.<br>3128. Repeat, building three more frames with the remaining 12-foot length of 2" X 4" lumber. |
| ***Document B: flowing text structure*** |
| If you have a yard that generates most any kind of green waste, you probably have the right ingredients and enough room to set up your own compost bin.<br>Composting is easy and cheap, you can cut down your garbage by hundreds of pounds each year, and create a mixture that can be used to improve the soil. |

The challenge is to suggest a reasonable way of measuring this feature, which is particularly important in a web search since lots of web pages contain point form information, links, labels or tags, titles, etc.

### 2.1 System design

Now that we briefly presented the criteria of knowledge-rich, specialized and flowing text structure, we go into more details about the corpus construction module design and the interaction between these criteria within that module. Figure 4 shows the corpus construction module conceptually divided in three steps. Step 1 addresses the flowing text structure criterion using a cleanness score. Step 2 combines the other two criteria of knowledge-rich and specialized documents in an iterative search. Step 3 is basically a decision unit controlling the stopping conditions for the corpus construction task. We describe each step in more details hereafter referring to the different blocks shown in the diagram of Figure 4.

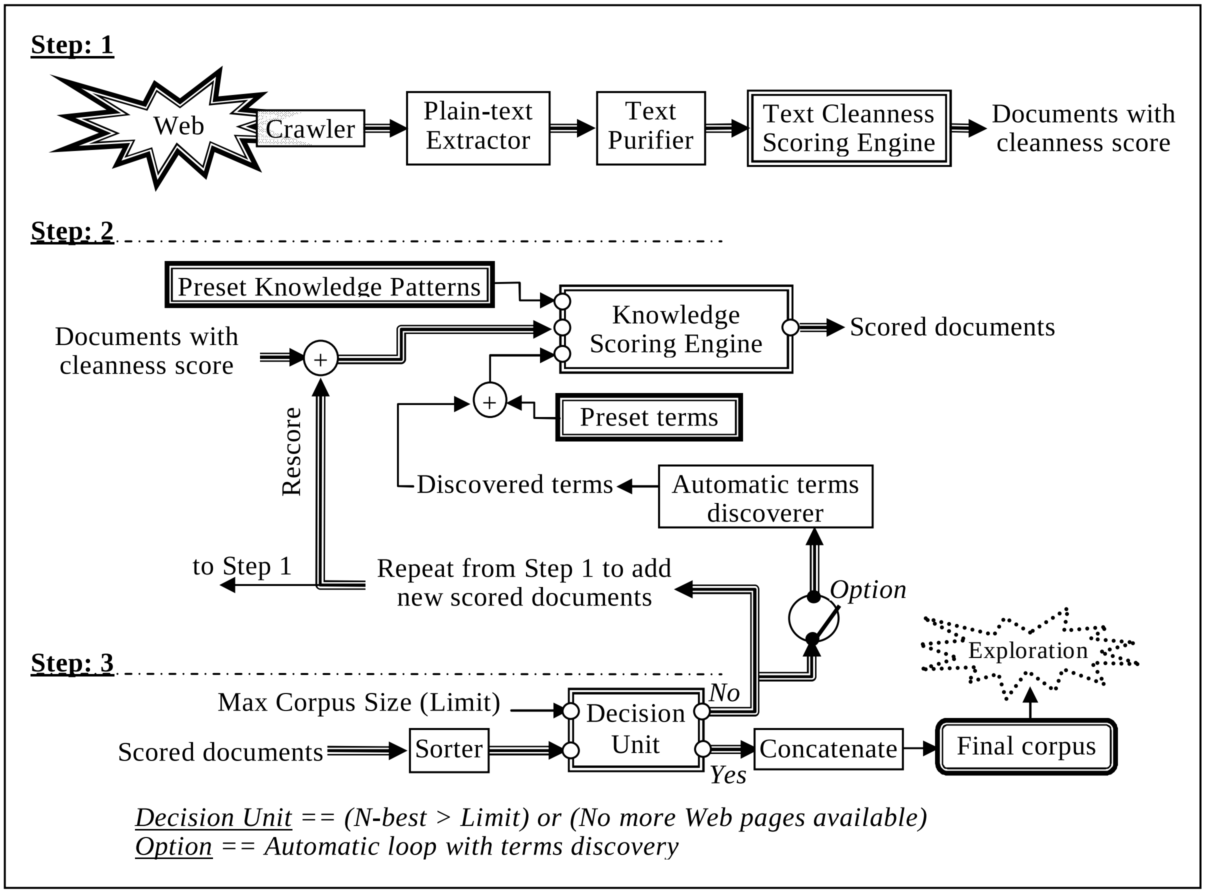

Step: 1
Web
Crawler
Plain-text Extractor
Text Purifier
Text Cleanness Scoring Engine
Documents with cleanness score
Step: 2
Preset Knowledge Patterns
Documents with cleanness score
Knowledge Scoring Engine
Scored documents
Preset terms
Rescore
Discovered terms
Automatic terms discoverer
to Step 1
Repeat from Step 1 to add new scored documents
Option
Exploration
Step: 3
Max Corpus Size (Limit)
Scored documents
Sorter
Decision Unit
No
Yes
Concatenate
Final corpus
Decision Unit == (N-best > Limit) or (No more Web pages available)
Option == Automatic loop with terms discovery

Figure 3: Detailed diagram of the Corpus construction module

Step 1

A Web crawler rides *Yahoo!* or *Google* search engines, using a given query, to collect documents (Html & PDF) from the Web and then extracts the plain-text content for each of them. Then, for each plain-text document, a Text Purifier suppresses all lines that are detected as “lexically poor”. The quantification of the poorness of a line is based on a parameter that we call “Line Average Cleanness” (LAC). LAC measures the quality of a line by counting how many lexical items it contains over all its tokens. Tokens can be anything such as words, punctuation marks, digits, or other tags. In fact, the cleanness depends on the characters allowed to form what we refer to as lexical items. We define three levels: VERY_CLEAN considers as lexical items tokens strickly made of alphanumeric letters, MERELY_CLEAN tolerates leading or ending punctuations, and POORLY_CLEAN consider any non white space separated chain of characters as a lexical item. The default LAC threshold for TerminoWeb is 70%[1] (any line with LAC smaller than 70% would be removed) which can be applied in combination with any of the 3 levels of cleanness to identify lexical items.

[1] Many of the default values mentioned in this article have been either set experimentally or arbitrarily. We assume such parameters can be changed by the user, or in future work we can investigate automatic approaches for finding optimal values for them.

Looking back at Table 2, LAC would return very low values for lines in Document A thus trigger their deletion from the extracted plain-text and high values for lines in Document B thus keep them. By removing lines from the extracted plain-text, we certainly loose data, but we do not think this is detrimental to the quality of the final result. On the contrary, since such lines do not bear interesting information for the terminologist, removing them is helping removing noise.

After the purification, we score the cleanness of the remaining text as shown in Equation 1 as a weighted sum (coefficients *a*, *b* & *c*) of the following evaluation features:

- Document Global Cleanness, to indicate the richness of a document in good lexical items.
- Document Richness in Discourse to estimate the quality of sentences in the corpus. A high value hints to a discursive document formed of long sentences and a very low value could point out very technical documents most likely made of titles, short mnemonic text lines followed by formulas or diagrams, URL links, etc.
- Document Size to possibly give relevance credits to long documents.

$$\text{Text Cleanness} = a * \left.\frac{\text{total lexical items}}{\text{total tokens}}\right|_{corpus} + b * \left.\frac{\text{total lexical items}}{\text{total Lines}}\right|_{corpus} + c * \text{CorpusSize} \quad (Equ.1)$$

At the end of this Step 1, we have plain-text documents with cleanness scores as estimates of their flowing text structure. At this point, the documents are individually available for reading, but we wish to further characterize them in terms of the value of their content (not only of their structure) for the terminologist. We continue to step 2.

Step 2

This step is to score newly extracted plain-text documents from Step 1 with respect to the knowledge-rich and specialized criteria. We measure the density of Knowledge Rich Contexts (KRC) in each document. A KRC is a sentence combining a Knowledge Pattern (KP) to consider the knowledge-rich criterion and one or two terms to consider the specialization criterion. The measurement uses a list of preset Knowledge Patterns (some examples were shown in Table 1) plus a list of some domain related terms. The system can start with a list of preset terms and then, depending on the system's configuration, dynamically update the list of terms with new terms automatically discovered from the input candidate documents during the processing. At the output of Step 2, we have plain-text documents characterized with the following scoring features: Text Cleanness (from Equation 1), Knowledge Patterns density (KP) and Knowledge Rich Contexts density (KRC). The densities of KPs and KRCs are measured relatively to the length of the document.

Step 3

A filtering process is carried out to rank the documents according to a weighted combination of their scoring features described in Step 1 and 2 as shown in Equation 2:

$$\text{Text Quality} = \alpha * \text{TextCleanness} + \beta * \text{KP} + \gamma * \text{KRC} \qquad (Equ.\ \mathbf{2})$$

Although $\alpha$, $\beta$ and $\gamma$ were respectively set to 15%, 25%, 60% in our experiments, these parameters values can be modified by the user. Ideally, the parameters should be automatically optimized if some reference corpora manually scored by experts were available.

## 2.2 Results

To provide the reader with a sense of what types of documents are retrieved when balancing our diverse criteria (default setting), we show in Table 3 a few examples of documents retrieved on three different themes: paragliding, computer storage and bank fraud. Giving control to the user, the corpus will be made of the subset of documents selected by the user.

Table 3: Examples of retrieved documents

| DOMAIN | WWW | DESCRIPTION |
|---|---|---|
| Paragliding | www.paragliding.com | Paragliding school – lots of FAQ. |
| | en.wikipedia.org/wiki/Paragliding | Entry on paragliding from Wikipedia. |
| | www.paragliding-lessons.com | School on paragliding. |
| Computer storage | en.wikipedia.org/wiki/Computer_storage | Entry on computer storage from Wikipedia encyclopedia. |
| | www.census.gov/prod/ec02/ ec0231i334112.pdf | Economic census on manufacturing computer storage. |
| | pedia.nodeworks.com/C/CO/COM/ Computer_storage | Entry on computer storage from Nodeworks encyclopedia. |
| Bank fraud | en.wikipedia.org/wiki/Bank_fraud | Entry on bank fraud from Wikipedia encyclopedia. |
| | www.usdoj.gov/criminal/fraud/idtheft.html | FAQ about Identity Theft and Fraud. |
| | www.quatloos.com/prime_bank_fraud.htm | Article from a non-profit corporation on financial and tax fraud education. |

## 3 Term extraction

The second main module of TerminoWeb is the term extractor. It is an important task in knowledge structuring to first find the important terms of a domain. We have already mentioned in Section 2 that the term extraction could be integrated in the iterative corpus construction module for finding Knowledge-Rich Contexts. However, it stands as a complete and fully featured autonomous module that could be used on any corpus, whether such corpus is built within TerminoWeb or not.

TerminoWeb's Term Extraction module relies on methodologies known in the literature to find single-word and multi-word terms. Two statistical approaches are used to find single-word terms. The first one uses *frequency*. It is an intra-corpus approach as we are simply measuring the frequency of each vocabulary word (excluding stop words) within the corpus. The second approach uses *weirdness* [5]. It is an inter-corpora approach as it relies on a comparison between a specialized corpus and a reference corpus. The British National Corpus (BNC) is used in this research as the reference general language corpus. Weirdness simply expresses the relative frequency of a word between the specialized and the general corpus.

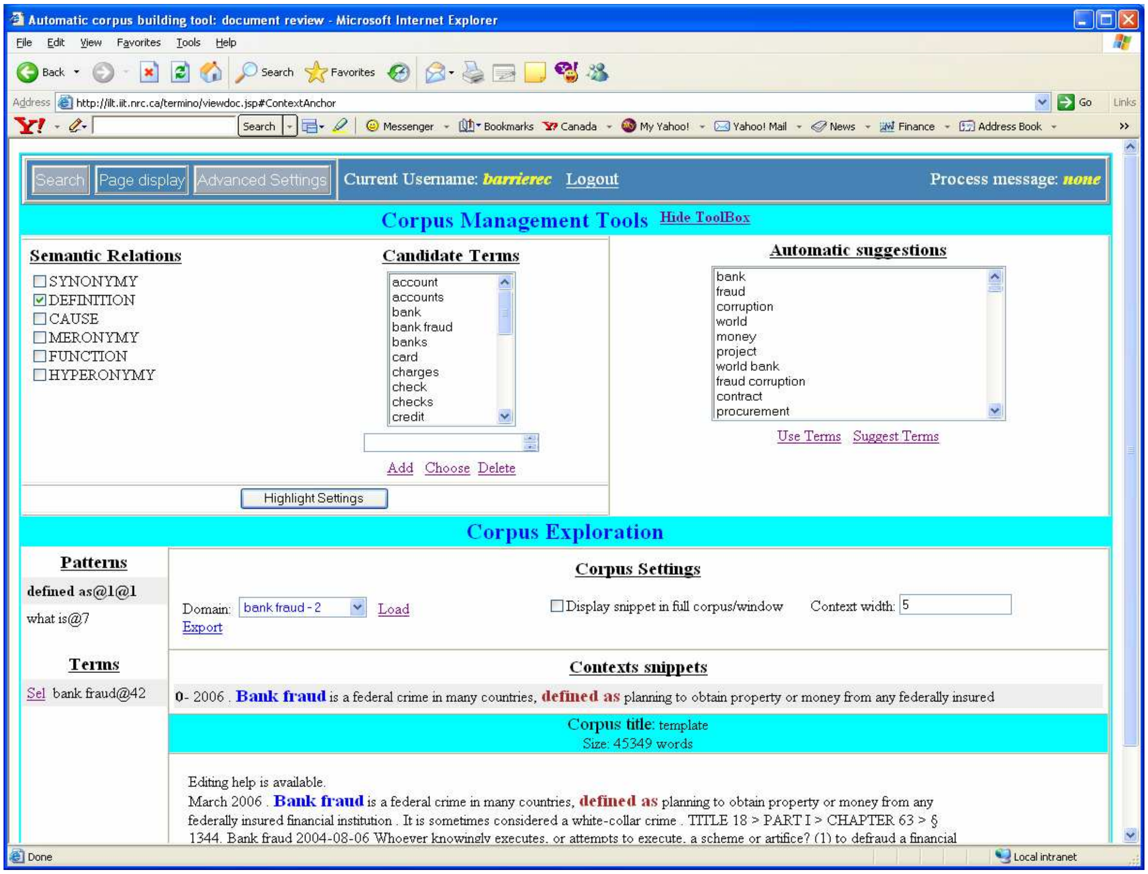
The Term Extraction module replicates Smadja's XTRACT model [6] to expand single-word terms into multi-word terms.

In Table 4, we revisit the three domains mentioned in section 2 for corpus construction: paragliding, computer storage and bank fraud. We provide the list of top 30 terms as found by using frequency for finding single-word terms and then performing multi-word expansion. The extraction is performed on a corpus of documents selected from the top 20 documents retrieved in the corpus construction module.

As we promote an interactive approach in TerminoWeb, we do not impose the list of terms resulting from the automatic extraction as the final list. Control is given to the user to transfer selected terms among the automatically suggested list to his/her validated terms list. More terms can also be manually added if needed. Figure 4 an interface in TerminoWeb where in the top half the user can select terms. The bottom half of the interface is for exploring a term candidate to find its occurrences in the corpus and hopefully find knowledge rich contexts around it. We present this idea in the next section.

Table 4: Examples of extracted terms

| DOMAIN | FREQUENCY |
|---|---|
| Paragliding | Paragliding, flying, paragliders, pilot, pilots, fly, air, paraglider, flight, wing, listing, manufacturers, site, video, hang, sport, book, training, retailers, dominican, gliding, launch, check, sites, equipment, hand gliding, flights, landing, republic, page |
| Computer storage | storage, computer, memory, information, disk, data, access, scsi, devices, hard, magnetic, furniture, office, media, computer storage, fibre, primary, channel, secondary, drive, tape, disks, drives, device, home, ata, computers, news, office furniture, fibre channel |
| Bank fraud | Bank, fraud, corruption, world, money, project, world bank, fraud corruption, contract, procurement, loan, financial, banks, identity, information, account, staff, fraudulent, credit, projects, theft, funds, borrower, business, number, card, investment, management, public, accounts |


Automatic corpus building tool: document review - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Messenger Bookmarks Canada My Yahoo! Yahoo! Mail News Finance Address Book
Search Page display Advanced Settings
Logout
Corpus Management Tools Hide ToolBox
Semantic Relations
Candidate Terms
Automatic suggestions
Use Terms Suggest Terms
Add Choose Delete
Highlight Settings
Corpus Exploration
Patterns
Terms
Corpus Settings
Load
Export
Contexts snippets
Done
Local intranet

http://ilt.iit.nrc.ca/termino/viewdoc.jsp#ContextAnchor

Current Username: *barrierec*

Process message: *none*

□ SYNONYMY
☑ DEFINITION
□ CAUSE
□ MERONYMY
□ FUNCTION
□ HYPERONYMY

Candidate Terms: account, accounts, bank, bank fraud, banks, card, charges, check, checks, credit

Automatic suggestions: bank, fraud, corruption, world, money, project, world bank, fraud corruption, contract, procurement

Patterns: defined as@1@1, what is@7

Terms: Sel bank fraud@42

Domain: bank fraud - 2

□ Display snippet in full corpus/window

Context width: 5

0- 2006 . **Bank fraud** is a federal crime in many countries, **defined as** planning to obtain property or money from any federally insured

Corpus title: template
Size: 45349 words

Editing help is available.
March 2006 . **Bank fraud** is a federal crime in many countries, **defined as** planning to obtain property or money from any federally insured financial institution . It is sometimes considered a white-collar crime . TITLE 18 > PART I > CHAPTER 63 > § 1344. Bank fraud 2004-08-06 Whoever knowingly executes, or attempts to execute, a scheme or artifice? (1) to defraud a financial

Figure 4: TerminoWeb: term extraction and knowledge-rich context search panel

## 4 Knowledge-rich contexts search

The third module of the overall system presented in Figure 1 is the search for definitional knowledge which can be used for a better understanding of the intrinsic properties of a term and of the interrelation of one term to other terms. We have introduced earlier our hypothesis that much definitional knowledge is given in text by knowledge-rich contexts indicative of semantic relations.

Presently, TerminoWeb offers the possibility to explore the occurrences of different terms, as shown in Figure 4. More particularly, its interesting exploration feature is to combine the search for a term to the search of a nearby Knowledge Pattern (KP), leading in fact to the search of specific KRCs. This is a good way of quickly leading to definitional knowledge. In Table 5, we revisit the same three domains presented earlier and show some definitional knowledge found. In each example, one term is highlighted in bold and the KP is underlined. Each KRC could contain more than one term.

Table 5: Examples of knowledge-rich contexts

| DOMAIN | KNOWLEDGE-RICH CONTEXT |
|---|---|
| Paragliding | A **paraglider** is a non-motorized, foot-launched inflatable wing. |
| | This rising air can come from two sources: when the sun heats features on the ground, columns of rising air known as **thermals** are generated when wind encounters a ridge in the landscape, the air is forced upwards, providing ridge lift. |
| | Techniques such as **no-wind reverse launches** and tandem launches are explained and demonstrated. |
| Computer storage | **Secondary storage**, also known as peripheral storage, is where the computer stores information that is not necessarily in current use. |
| | Hard drives, CD-ROMs, **tape drives**, and other fixed-connectivity devices have little life past their original topology. |
| | **Formatting** is also called initializing. |
| Bank fraud | **Bank fraud** is a federal crime in many countries, defined as planning to obtain property or money from any federally insured financial institution. |
| | Your personal data especially your Social Security number, your **bank account** or credit card number, your telephone calling card number…. |
| | A **booster cheque** is a fraudulent or bad cheque used to make a payment to a credit card account in order to "bust out" or raise the amount of available credit on otherwise-legitimate credit cards. |

## 5 Conclusions and Discussion

We have presented the three important modules currently integrated into the TerminoWeb platform, a software environment for terminologists. We have described each module: corpus construction, term extraction and knowledge-rich contexts exploration. Obviously, much more work can be done in such project, as each of the modules can be further developed, and the theoretical ideas behind the different application modules can be further explored.

For example, in the corpus construction module, we need to better assess the impact of different parameters in the equation leading to the document quality scoring. Empirical data would be valuable here to determnie what in practice is the best balance of parameter values for a good estimation of text quality. One important aspect which we are currently investigating is the impact of different noise levels on the different Knowledge Patterns (KPs) on the corpus building model.

Although they often express a semantic relation, many KPs express other things also. For example "is a part of" is unambiguous in its indication of a meronymy relation, but "is a" although a very good hyperonymy pattern, is also a very noisy one. Therefore a naïve count of KP occurrences in documents as we do in the present implementation of TerminoWeb without a disambiguation module, certainly leads to some errors in the ranking of the documents.

The term extraction module can be studied as an independent entity. So far, we have implemented approaches documented in the literature in computational terminology, but we are looking into other novel approaches as well. Evaluation of these approaches is often problematic and needs further investigation.

The knowledge-rich contexts exploration module also suffers from not having a KP disambiguation algorithm included to determine when a pattern actually means a semantic relation or not. Much more work toward an automatisation of this module can be envisaged. This ambitious path could eventually lead to automatic knowledge structuring to help presenting to the terminologist an organized network of terms rather than a simple list.

## References

[1] L'Homme, M.C. (2004) *La terminologie: principes et techniques.* Les Presses de l'Université de Montréal.

[2] Meyer, I. (2001), Extracting Knowledge-rich Contexts for Terminography: A Conceptual and Methodological Framework. In D. Bourigault, C. Jacquemin and M.C. L'Homme (Eds), *Recent Advances in Computational Terminology*, 279-302, John Benjamins.

[3] Barrière, C. (2004) Knowledge-Rich Contexts Discovery. *Proceedings of the 17th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence,* Canadian AI 2004, London, Ontario, Canada, May 17-19, 2004.

[4] Agbago, A. & Barrière, C. (2005) Corpus Construction for Terminology, In P. Danielsson and M. Wagenmakers (eds) *Proceedings from the Corpus Linguistics Conference Series*, vol. 1, no.1, available online at http://www.corpus.bham.ac.uk/PCLC/.

[5] Lee G., Mariam T., and Khurshid A. "Terminology and the construction of ontology", Ibekwe-SanJuan, Fidelia, Anne Condamines and M. Teresa Cabré Castellví (eds.), *Application-Driven Terminology Engineering: Special issue of Terminology* 11:1 (2005). 2005. 231 pp. (pp. 55–81)

[6] Smadja, F. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143-177