Corpus Construction for Terminology

Agbago, Akakpo; Barrière, Caroline

Publisher's version / Version de l'éditeur:

Corpus Linguistics 2005 Conference [Proceedings], 2005

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

Corpus Construction for Terminology *

Agbago, A., and Barrière, C.
July 2005

* published at the Corpus Linguistics 2005 Conference. Birmingham, United Kingdom. July 14-17, 2005. NRC 48516.

Canada

CORPUS CONSTRUCTION FOR TERMINOLOGY

Akakpo Agbago and Caroline Barrière
Interactive Language Technology Group, Institute for Information Technology,
National Research Council of Canada
{akakpo.agbago,caroline.barriere}@nrc-cnrc.gc.ca

ABSTRACT

In this research, our theoretical goal is to investigate what characterizes relevant documents for use in terminological work, and our practical goal is to develop a web-application to help terminologists in their task of building a domain-specific corpus. Meyer (2001) defines knowledge patterns as guides for discovering knowledge-rich contexts which embed semantic relations between terms. Inspired from Meyer's work, our contribution is to suggest a "knowledge-richness" estimator to evaluate the usefulness of a text based on its density of knowledge patterns. We evaluate this hypothesis and present some results. We further use this estimator in combination with a search engine on the Web for document ranking. We present the corpus construction and management tool and some results.

1. Introduction

Terminologists, browsing through texts about a specific domain, must be able to understand the important concepts and semantic relations of that domain to further structure its information in a concise way. Texts on any domain are today easily available on the Web. The problem is not availability but mostly quality or even just usefulness of the information for the purpose of understanding a domain.

In this research, our theoretical goal is to investigate what characterizes relevant documents from a terminological point of view, and our practical goal is to develop a web-application to help terminologists in their task of building a domain-specific corpus. Although different guidelines have been given in the literature (L'Homme 2004) for terminologists to determine the relevance of a document as to be included or not in a corpus, most of these guidelines are difficult to measure quantitatively. Since we aim at the automatisation of the corpus building process, we look for a way to characterize a text which is measurable and which provides an appreciation of its value. This leads us to go one step further and ask "What will the terminologists look for in the texts *after* the corpus is built?" Then, we develop our strategy from the answer to that question.

In fact, much corpus analysis in terminology aims at first, finding important terms in the domain, and second, at finding knowledge patterns (Meyer 2001) indicative of semantic relations between these terms. For example, *such as, is another, is a kind of,* are all knowledge patterns showing a hyperonym relation. The list of terms is not known in advance, but the list of knowledge patterns is, at least partially. Certainly not all domains express their semantic relations using exactly the same knowledge patterns (Meyer et al. 1999). There is some variation, but a basic set of knowledge patterns can be listed as used across domains. The density of occurrence of any knowledge pattern from this list in a text is therefore a measurable feature. Our contribution in this research is to suggest and validate the hypothesis that a "knowledge-richness" estimator can evaluate the usefulness of a text based on its density of knowledge patterns. Our hypothesis is that the texts chosen by terminologists will have a higher density of knowledge patterns than randomly chosen texts. We elaborate on knowledge patterns and knowledge-rich contexts in

section 2 to present our hypothesis and then present an experiment to validate this hypothesis in section 3.

In section 4, we will then present a corpus management tool, called TerminoWeb, which uses the knowledge-richness estimator in combination with a web search engine to retrieve useful documents from the web. The tool is very flexible, allowing a user to construct multiple domain-specific corpora, and obtain for each one a set of web documents sorted by decreasing value of knowledge-richness. The terminologist can then view and select the documents to be included in each domain-specific corpus built. To help that decision, corpus analysis is also performed to highlight the knowledge patterns, or any other user-defined pattern in the text.

As we conclude in section 5, we show that overall our approach allows the system to learn which texts are valuable to a terminologist and to increase its performance over time to give an accurate knowledge-richness characterization of a text.

2. Our hypothesis: knowledge patterns can help find "good" texts

Most terminological work assumes a manually created corpus before involving the use of any tool to help the terminologist toward the construction of a Terminological Knowledge Base (TKB). The corpus construction step is a critical one, as the terminologist must retrieve domain-specific texts from different sources. These texts should not be any texts. In fact, the problem is not finding texts, it is finding valuable texts. Terminologists must often look through many texts before finding appropriate ones. There are guidelines for choosing them, as is presented in L'Homme (2004, p. 126ff), to be qualitatively measured about a text such domain specificity (how much the text corresponds to the domain of interest?), language (text can be selected from all languages as one important task in terminology is to define equivalences), *originality (*texts should not be translations), *specialization level*: the difficulty of the text whether it's written for experts or general audience, *Type*: the style of the literature (scientific, pedagogical, business), d*ate*: recent or deprecated subjects, *Data evaluation*: authors or publisher's reputation.

Although these guidelines help choosing, few of them can be performed automatically, such as date retrieval and author retrieval. The terminologists must still go through many texts to decide which one to keep.

Looking more specifically at the specialization level criteria, Pearson (1998: 60-61) mentions that the types of texts which tend to give term explanation and relation to each other are texts which assume an expert-to-novice communicational goal as opposed to expert-to-expert in which much information can remain implicit. An expert-to-novice text will tend to render all new notions explicit to ensure the understanding by the reader. Texts written with a communicative goal of information (even popularization) will contain much semantic relations between concepts (synonymy, hyperonymy, meronymy) expressed explicitly.

The explicit expression in text of semantic relations between concepts is often via specific surface patterns. Meyer et al. (1999) has called them Knowledge Patterns and her work provides much insight on their definitions and their use within a terminological context. Following in that direction, Barrière (2004) presents an extensive study of knowledge patterns, looking at their presence in corpora as well as in electronic dictionaries.

The concept of Knowledge-Rich Contexts (KRC) was introduced also in Meyer et al. (1999) as sentences of interest to terminologists because they embed both important domain terms and

knowledge patterns. The relationship of knowledge patterns and terms is that they help to better understand the conceptual relations in which the terms stand.

So in the quest of computable means to measure the richness in knowledge of documents, we make the concept described above our fundamental hypothesis. The advantage of computable measures is that we can further suggest an automatic process to build a corpus (see section 4). A "good" text will be rich in explicit semantic relations, particularly paradigmatic relations. The richness of the text will be even greater for the terminologist if the knowledge patterns are involved in simple semantic contexts of value in the domain studied. For example, a sentence such as *compost is made of fungi humus soil* within a corpus on composting, contains *is made of* as the knowledge pattern, *compost* and *soil* as domain-specific terms. The presence of the terms around the knowledge pattern certainly increases its value.

We define a KRC as follow:

term * Knowledge-Pattern * term (1)

In words, the expression above means:

- given a certain length of text,
- we look for the presence of a term in the same context as a knowledge pattern,
- the presence of such term can be either at left or right of the knowledge pattern or both,
- the "*" is a wildcard used to represent a limited number of other words allowed in between. It fixes the length of the context (window size) in term of number of words.

The question that comes up with the measurement of the KRC is how do we know the terms to begin with? If the purpose of building a corpus is to eventually perform term extraction to find the terms, we cannot use these terms to validate the value of a text. This is the chicken-and-egg problem. We will discuss in section 5, how to implement a system to iteratively be able to measure more and more precisely KRC density.

But for now, let us say that the expression (1) above leads to two different measures, (a) KP - knowledge pattern density (assuming we do not know the terms) and (b) KRC – knowledge-rich context density (knowing which terms are important).

3. Experimentation for hypothesis validation

To validate our hypothesis, we compare two corpora built by experts in terminology respectively in the domains of *Scuba diving* and *composting*, with two corpora on the same domain made of web documents found by a search engine. (The corpora from experts were made available to us by late Ingrid Meyer, professor at School of Translation and Interpretation, at the University of Ottawa). There are many free and popular search engines that can browse the web and come up with documents relevant to a specific domain. However, there is no guarantee that the documents contain explicit semantic relation contexts that are very important and useful to describe a domain. Such documents are indeed domain-specific but they might knowledge-poor as opposed to knowledge-rich.

The main two inputs to our system are first the query term for the search operation and second the set of semantic relation knowledge patterns. The query term is a term central to the domain that would be used for initiating the corpus construction process. The set of knowledge patterns is collected from the literature in terminology on knowledge extraction (Barrière 2004). The list of 75 knowledge patterns is grouped into 6 semantic relation types: Synonymy, Hyperonymy,

Meronymy, Definition, Function and Cause. Table 1 shows our query terms, and Table 2 shows a few knowledge patterns, and the complete list is put in Appendix 1.

Corpus domain	Query Term
Scuba diving	"scuba diving"
Composting	"compost"

Table 1 – Query terms

Semantic relation	Pattern examples
Synonymy	is another word for, also known as, also called
Hyperonymy	is a kind of, is classified as, is a sort of
Meronymy	is composed of, is a part of, is a component of
Definition	is defined as
Function	is a tool for, is made to, is designed for
Cause	influence, promote, lead to, prevent

Table 2 – Examples of knowledge patterns

The first results show an important difference between the terminologist's corpus (referred to as Baseline) and Google corpus, as shown in Table 3. Each Google corpus was made by using the query term in Table 1 to launch the search engine, and then taking the top X documents from Google Web APIs (beta) and concatenating them to obtain a corpus of comparable size to the human corpus. The percentages reported in Table 3 correspond to the total number of occurrences of knowledge patterns (KP) divided by the number of words in the corpus. The last column (difference) shows the relative proportion of patterns in Google with respect to the terminologist's corpus, as calculated by (Google – Baseline)/ Baseline.

Title	Size in words	KP Density	Difference
Compost Baseline	88165	1.27602%	-15.37%
Compost Google	88166	1.07978%	
Scuba Baseline	134253	0.86702%	-39.67%
Scuba Google	134408	0.52303%	

Table 3 – Comparing Google to terminologist's corpus (Baseline) as to their knowledge pattern density

Now, let us go one step further. Knowledge patterns can be noisy, meaning that their presence in text does not necessarily lead to a knowledge-rich context. Let us take an example for a function relation with the pattern *used to*. In a sentence such as *drug like those are* ***used to*** *control cold symptoms* we definitely have a function relation, but, in a sentence such as ***I used to*** *go so far as to tell people,* it is certainly not. The negative example shows a case that would be wrongly counted as a knowledge pattern thus wrongly increasing the score for a document.

Unfortunately, disambiguation of the meaning of patterns is not an easy task, even though different linguistic mechanisms could be put in place for such disambiguation, such as syntactic or semantic analysis. As those are complex (and would introduce much delay in a search system), we opt for a more "terminology" approach, as we view the value of knowledge patterns increase if they are in presence of terms of interest for the domain to be studied.

Of course, in normal circumstances (during system usage), terms would not be known before hand as we mentioned previously. However, for the sake of our experiments toward validation of our hypothesis, let us assume a scenario in which we have come to a level of iteration where we have become familiar with some terms in the domain. In the case of this evaluation, it is equivalent to deriving these terms from the terminologist's corpora which we are using as

Baseline. To do this task, we use a term extractor called "TermoStat Web" (available online at: http://olst.ling.umontreal.ca/~drouinp/termostat_web/ and based on principles described in Drouin 2003). Table 4 shows the top 20 terms extracted for both corpora.

Corpus	Top 20 terms
Scuba	dive, underwater, immersion, waterproof, oxygen, dive, mask, cave, oxygen, cavern, instructor, regulator, symptoms, depth, nitrogen, feet, wreck, underwater, nitrogen narcosis, air, cave diving, buddy, surface, snorkel, boat
Composting	Pile, materials, soil, nitrogen, bin, compost pile, organic materials, bacteria, worms, leaves, leaf, decomposition, organisms, temperature, process, ratio, carbon, nutrients, organic matter, moisture, grass clippings

Table 4 – Terms extracted by TermoStat

If we go back to Expression (1) given earlier in chapter 2, the results in Table 3 were for knowledge patterns and therefore they assumed that the terms on the left and right of the knowledge pattern could be anything. Now we present again in Table 5 the results of the same test but this time, we consider and count an occurrence of a knowledge pattern as valid only if it is part of a knowledge-rich context (KRC). This means that there must be within 10 words maximum either on the left, or on the right, or both of the knowledge pattern, at least one term of the list of 150 terms (top 20 of which is listed in Table 4).

Title	KRC Density	Difference
Compost Baseline	1.02535%	-28.76%
Compost Google	0.73044%	
Scuba Baseline	0.36722%	-.22.60%
Scuba Google	0.28421%	

Table 5 – Comparing Google to terminologist's corpus (Baseline) as to their Knowledge Rich Context density

There certainly still is an important difference between the two metrics. However, they show that the Baseline corpus is richer in KRC than Google built one. This shows support for our hypothesis. Although since that difference is in one case greater (for Compost) and in one case lesser (for Scuba), as we can see in Table 6, we do not conclude as to the impact of noise, and leave it to future work to look into such differences on a larger number of corpora. Nonetheless, the interpretation of the results for Scuba is that, even though KP metric sees Google corpus as very poor compared to the baseline, KRC metric finds that these few knowledge patterns of Google corpus are rather greatly involved in rich contexts. In fact, 57.64% of knowledge patterns in Scuba baseline corpus are not in rich contexts compared to only 19.64 % of Compost baseline corpus (the two values are calculated as KRC – KP over KP with data from Table 3 & 5).

Corpus	Difference in KP	Difference in KRC
Compost	-15.37%	-28.76%
Scuba	-39.67%	-.22.60%

Table 6 – Differences in KP versus KRC

Therefore, we can conclude that each metric, taken independently, provides a good characterization of a terminologist's corpus. The comparison with Google, highlights the high density of such patterns in the terminologist's corpora, as compared to the top X documents returned by Google. These results encourage toward the development of a Corpus Management Platform, to retrieve directly from the Web the most interesting documents for a terminologists,

"interesting" as we characterize it here in terms of something measurable (and therefore can be automated).

4. TerminoWeb

We developed TerminoWeb, a corpus-building web application that allows a terminologist to perform search for documents about particular domains and manage the results efficiently and consistently over different work sessions. Each user has a personalized working environment handled as an account. This aspect is useful for a terminologist who certainly needs to do some iterative work to refine his work as he gets new ideas. The user can define his own parameters, build his own corpora and update his work. We presented an earlier version of this environment in Barrière (2005). Hereafter we give details of the types of input it uses and output it generates, as well as key ideas for the design and results as to density of knowledge patterns and knowledge-rich contexts it provides to the texts it retrieves.

4.1 Inputs

In the formulation of search inputs, the user can define:
(1) a corpus domain
(2) a query-term
(3) a list of knowledge patterns
(4) a list of domain terms (if known)

The corpus domain is the principal domain for which we want to build the corpus (e.g. composting). It can be anything useful to the user to differentiate the corpus. The query term is a central term to the studied domain used to launch the search engine (Google API). For the knowledge patterns, a pre-selected list is given (the one presented in Appendix 1), but the user is free to search for specific subsets of this list by selecting only some semantic relations and not others, and can even further select a subset of patterns within a semantic relation. Furthermore, a user can create a list of new knowledge patterns of his choice.

The list of domain terms is the most tricky to have, and the system will work fine without it, as it will be able to provide knowledge pattern counts but not knowledge-rich context counts. An existing list of terms from a domain can be found in a term bank and used as a starting point. But maybe that list does not exist (the terminologist is trying to find the terms in a new domain) and therefore it will be empty. The purpose of corpus building is certainly to create from scratch or to add to an existing list of terms.

In Figure 1, we show the user interface which allows users to define new knowledge patterns. In Figure 2, we show the user interface to select the patterns to be used in a particular search. The same interface is used to give a query word, and select the corpus domain.

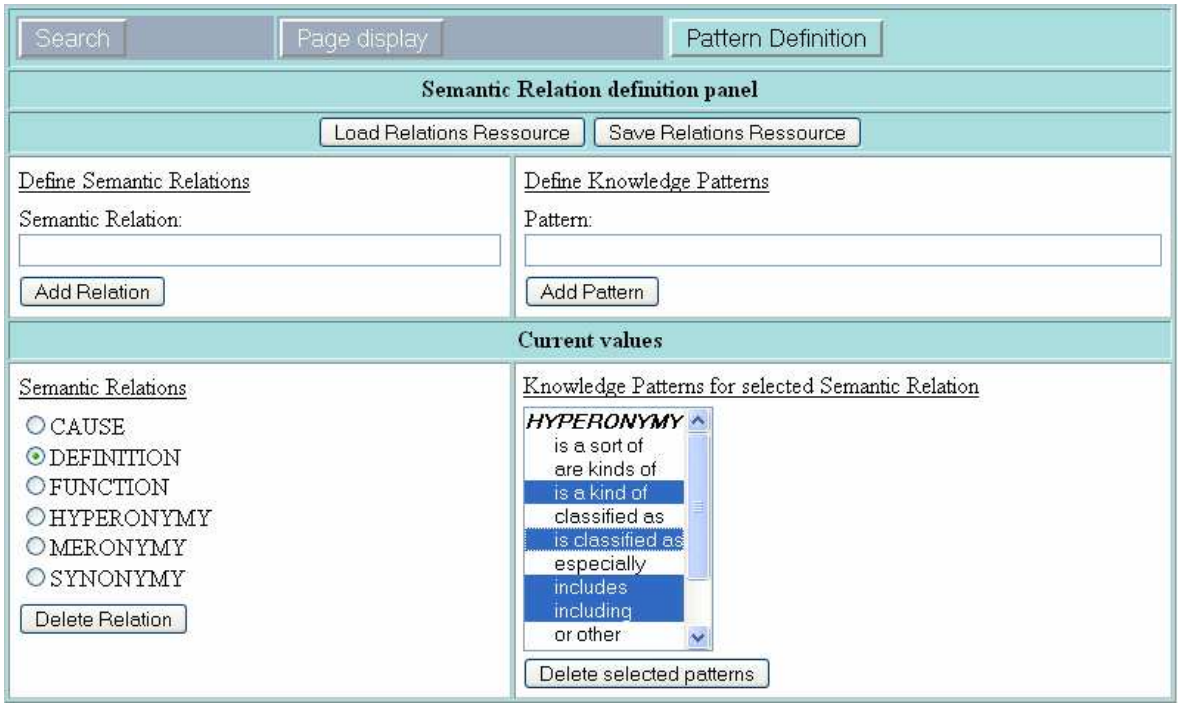


Figure 1: Knowledge patterns definition interface

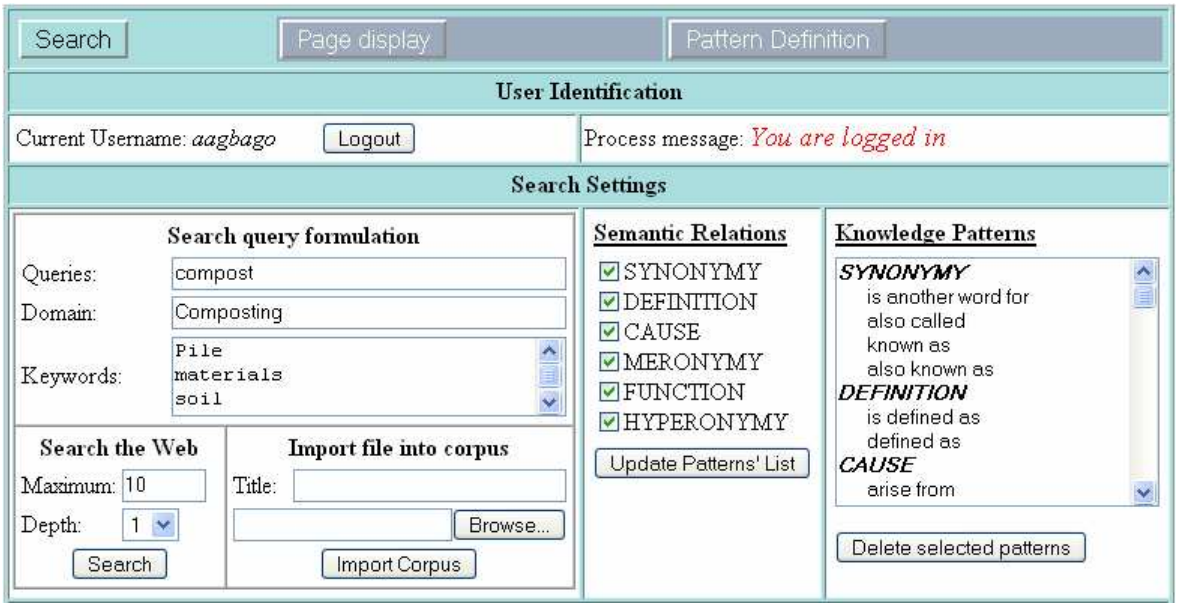
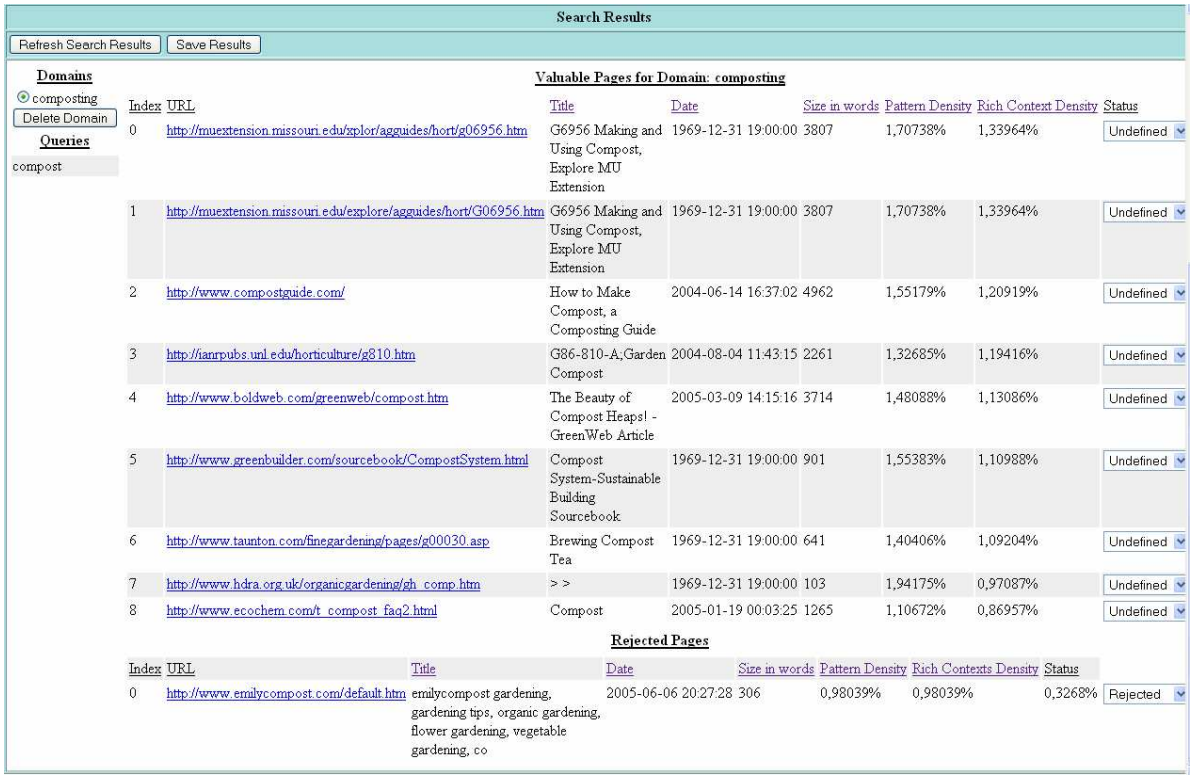


Figure 2: Search queries formulation interface with knowledge patterns selection

4.2 Outputs

TerminoWeb displays the resulting documents on a web page interface that allows the user to sort the list based on Knowledge-rich Context density (KRC), Knowledge Pattern density (KP) and other parameters. The main particularity of TerminoWeb is the display of search results, as

shown in Figure 3. The terminologist can access each document (as a link to the web page is provided) and then decide to accept or reject it in the status field.



Search Results

Refresh Search Results | Save Results

Domains

composting

Delete Domain

Queries

compost

Valuable Pages for Domain: composting

Index	URL	Title	Date	Size in words	Pattern Density	Rich Context Density	Status
0	http://muextension.missouri.edu/xplor/agguides/hort/g06956.htm	G6956 Making and Using Compost, Explore MU Extension	1969-12-31 19:00:00	3807	1,70738%	1,33964%	Undefined
1	http://muextension.missouri.edu/explore/agguides/hort/G06956.htm	G6956 Making and Using Compost, Explore MU Extension	1969-12-31 19:00:00	3807	1,70738%	1,33964%	Undefined
2	http://www.compostguide.com/	How to Make Compost, a Composting Guide	2004-06-14 16:37:02	4962	1,55179%	1,20919%	Undefined
3	http://ianrpubs.unl.edu/horticulture/g810.htm	G86-810-A;Garden Compost	2004-08-04 11:43:15	2261	1,32685%	1,19416%	Undefined
4	http://www.boldweb.com/greenweb/compost.htm	The Beauty of Compost Heaps! - GreenWeb Article	2005-03-09 14:15:16	3714	1,48088%	1,13086%	Undefined
5	http://www.greenbuilder.com/sourcebook/CompostSystem.html	Compost System-Sustainable Building Sourcebook	1969-12-31 19:00:00	901	1,55383%	1,10988%	Undefined
6	http://www.taunton.com/finegardening/pages/g00030.asp	Brewing Compost Tea	1969-12-31 19:00:00	641	1,40406%	1,09204%	Undefined
7	http://www.hdra.org.uk/organicgardening/gh_comp.htm	> >	1969-12-31 19:00:00	103	1,94175%	0,97087%	Undefined
8	http://www.ecochem.com/t_compost_faq2.html	Compost	2005-01-19 00:03:25	1265	1,10672%	0,86957%	Undefined

Rejected Pages

Index	URL	Title	Date	Size in words	Pattern Density	Rich Contexts Density	Status
0	http://www.emilycompost.com/default.htm	emilycompost gardening, gardening tips, organic gardening, flower gardening, vegetable gardening, co	2005-06-06 20:27:28	306	0,98039%	0,98039%	0,3268% Rejected

Figure 3: Results from search with provided ranking

4.3 Design

TerminoWeb embeds two engines: a search engine and a filter. The storage of users' accounts is handled by a database server.

- The search engine is embryonic. It rides Google search engine through an API to get entry point urls to the web and crawl on its own the downlinks of those primary pages from Google. This is to take advantage of the efficiency Google has acquired over the years. As Google API returns very few pages (10 maximum at a time) the crawling of the downlinks increases the recall of the search for the next step that is the filter.

- Filter: The essence of TerminoWeb depends on this algorithm because the richness of the filtered documents depends upon it. This is what makes the difference between TerminoWeb and other search engines. The challenge here is to make the documents returned more interesting than Google's. The filtering is solely based on Knowledge-Rich-Contexts and Knowledge Patterns.

4.4 Iterativity

Indeed, the system can not evaluate KRCs when the terminologist begins to create a corpus. However, it can evaluate knowledge patterns and use this to retrieve documents rich in patterns,

among which the terminologist will select some to become part of the corpus. On this partial corpus, we can perform term extraction to start creating our list that will be further refined as we find more texts.

Let us consider a typical scenario where a terminologist would like to build a corpus on scuba diving. His starting point, his first query word is *scuba diving*. In the first iteration, TerminoWeb would return documents where there will be plenty of knowledge patterns but not necessarily surrounded by domain terms. Using the resulting documents, the user might discover the keywords *dive, underwater, oxygen, boat, octopus*, and many others. He can then run a second iteration with the same word *scuba diving* as query, but now the keywords he has identified will be used to calculate KRCs and to optimize the list of documents returned. After a few iterations, our system would gather only the richest documents semantically thus allow the creation of a corpus that is interesting for a terminologist.

In the present version of TerminoWeb, the term extraction must be done manually (or a third party toolkit such as TermoStat). Future work will lead us to integrate an automatic term extraction module to perform the iterative process in the background, not having to burden the user with that step and providing better and better texts to him.

4.5 Results

With TerminoWeb in action, our expectation is that a corpus built with it (let us call it TerminoCorpus) would contain more occurrences of knowledge patterns and knowledge-rich contexts than the one built from the concatenation of the first top documents return by Google search engine (let us call it GoogleCorpus).

We revisit the domains of composting and scubadiving, as we have results for those from human terminologists (baseline). For each baseline corpus, we build 4 corpora: two from our system and two from Google. The first Google corpora (Google in Table 7) is built using the query term only for the search (see Table 1). The second Google corpora (OR query Google in Table 7) is built using a list of 20 domain terms (top 20 from TermoStat) with a logic OR query i.e. “compost OR dive OR …” This second corpus is mostly to give a second reference point with a corpus that is already very domain specific (as it is return by Google).

The two TerminoWeb corpora are built using TerminoWeb (our engine), which as described in the design section 4.3 would reorder the Google ranking to obtain at the top of the list, the texts with the highest density of Knowledge Rich Contexts. Shown in Table 7, the TerminoWeb 0 is where a KRC is actually reduced to a KP, since no context is taken into consideration, and TerminoWeb 3 resp. 10 is using a window of 3 resp. 10 words to the left and right of the KP to find a term contained in the list of 150 terms per domain extracted by TermoStat. We note that the windows are set for the filtering process while we collect documents from the Web. That is where it has an impact on the ranking of the documents, thus as to what documents goes into the resulting corpus. The case of TerminoWeb 0 means that only the first richest documents in KP not KRC are used for the resulting corpus. But the results in Table 7 are evaluations on the same 10 words window scale on those different corpora. To better see the difference between all corpora built, we use the baseline as the comparison point, and provide relative densities calculated as (density of X – density of Baseline) / density of Baseline.

Title	Size in words	Pattern Density	Rich Context Density
Compost TerminoWeb 3	88165	32.80%	38.49%

Compost TerminoWeb 10	88165	34.31%	38.16%
Compost 20 OR query Google	88165	18.75%	16.70%
Compost Baseline	88165	0	0
Compost TerminoWeb 0	89399	59.36%	-3.34%
Compost Google	88166	-15.37%	-28.76%
Scuba TerminoWeb 10	134292	11.82%	79.25%
Scuba TerminoWeb 3	134252	4.55%	65.92%
Scuba TerminoWeb 0	134884	54.94%	43.54%
Scuba Baseline	134253	0	0
Scuba 20 OR query Google	135432	-40.04%	-20.77%
Scuba Google	134408	-39.67%	-.22.60%

Table 7 – Evaluating TerminoWeb

Results in Table 7 show that our system:

- Succeeds to create not only a corpus rich in KP but with the highest KRC possible
- Outperforms Google and even Google with multiple terms in KP and KRC.

4.6 Other features of TerminoWeb

(1) Other measurable criteria

Two other easily measurable features have been included in TerminoWeb: the size of the text and the date. The date might be interesting for the studying of the term in different time to see its evolution over time for example.

(2) Corpus Management

Since users might already have texts for which they want to calculate the KP and KRC, we provide a way to upload these files. As mentioned earlier, a terminologist can accept or reject a file to be part of a corpus. We keep track of those decisions. We also keep track of the different query terms used for one domain. We allow different corpus on different domains to be created for a single user. The web interface can give access to multiple users simultaneously.

(3) Text exploration

It is possible for the user to open the web page to directly have access to its original content. But also, we provide a pattern search module for the user to see within a text the occurrences of knowledge patterns (Figure 4).

Figure 4: Looking inside the documents

5. Conclusions and future work

In this research, we stated a hypothesis of a measurable criteria for characterizing a corpus made by terminologists, that of its density of knowledge patterns, and even of its knowledge rich contexts. We suggested an experimental protocol to validate our hypothesis which consisted in comparing two terminologist-made corpora with two corpora on the same topic made by gathering texts using a well-known search engine, Google. We showed that both knowledge patterns and knowledge-rich contexts had a higher density in the terminologist corpus.

Certainly, it is difficult to provide generalization from results on two corpora. The study of more corpora on different domains should be done to provide a more in depth evaluation of our hypothesis. Furthermore, this study assumed that all knowledge patterns were of equal value, not differentiating between the different semantic relations. In future work, we will perform a more in depth analysis of the contribution of each type of semantic relation to the value of a text.

We showed the development of TerminoWeb, a corpus construction and management tool for terminologists. The main strength of TerminoWeb is to provide a reordering of a standard search engine (Google) with respect to knowledge pattern and knowledge rich context density.

Future mostly lies, as mentioned earlier, in adding a term extraction module to provide better knowledge rich context evaluation. We envisage an adaptable system, which between the beginning state (with no terms known) and the final state (with all terms extracted), provides a weighting of KP and KRC results, giving less and less weight to the KP and more and more to the KRC in a series of iterations.

We presented TerminoWeb's other features of characterizing the text with the date and size on top of KP and KRC. A future research direction is to investigate the list of features given in L'Homme (2004) and find which ones could be partially automatize to provide an even better characterization of each text retrieved on the Web.

References

Barrière, C. (2004) Knowledge-Rich Contexts Discovery. 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.

Barrière, C. (2005) Semi-automatic corpus construction from informative texts. In L. Bowker (ed), Text-based Studies. Lexicography, Terminology, Translation. In Honour of Ingrid Meyer. University of Ottawa Press. (to be published)

Drouin, Patrick (2003). Term extraction using non-technical corpora as a point of leverage. In *Terminology*, vol. 9, no 1, p. 99-117.

Google Web APIs (beta). Available at http://www.google.ca/intl/en/apis/ (assessed June 13th, 2005)

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

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.

Meyer, I., Mackintosh, K., Barrière, C., and Morgan T. (1999), Conceptual sampling for terminographical corpus analysis. In: *Terminology and Knowledge Engineering, TKE'99*, Innsbruck, Austria, 256-267.

Pearson, J. (1998) "Terms in Context", John Benjamins Publishing

Appendix 1 – Knowledge patterns used in experimentation

Hyperonymy	such as and other or other including includes especially is classified as classified as is a kind of are kinds of is a sort of are sorts of
Synonymy	known as also known as also called is another word for
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
Definition	defined as is defined as
Function	is needed is designed for is made for is made to is essential to functions of in order to is a tool to

Cause	ensure affect help influence play a role in yield contribute allow permit makes a difference provide result generate create produce enhance increase improve aid promote eliminate reduce kill finish put an end to stop destroy deter decrease discourage prevent maintain lead to cause make to achieve need depends on arise from