The Role of Technical Terminology in Question Answering

Fabio Rinaldi, James Dowdall, Michael Hess,
Kaarel Kaljurand, Magnus Karlsson

Institute of Computational Linguistics, University of Zürich
Winterthurerstrasse 190, CH-8057 Zürich, Switzerland
{rinaldi,dowdall,hess,kalju,karlsson}@cl.unizh.ch

Résumé

Terminology is arguably the most vital linguistic unit of technical documentation. Characterising the content of documents by the terminology they contain is a key factor in satisfactory document retrieval. But when users require answers rather than documents, more complex strategies for exploiting terminology are needed.

Dealing effectively with this problem requires not only good techniques for terminology extraction but also ways to organize and structure the terminology. We describe some potential solutions to this problem, taking a Question Answering system as an example. We show which benefits our techniques bring to the system.

1. Introduction

The pivotal role terminology plays in technical domains has long been recognized. Whilst terminology extraction methodologies have received much attention, strategies of exploiting this knowledge persistently revolve around a common theme - shallow processing to produce domain descriptions or knowledge bases. For good reason, as complex multi-word terms quickly become a thorn in the side of computational accuracy and efficiency when a deeper linguistic analysis is required.

One area where such requirements are particularly pressing is that of 'Question Answering' (**QA**), in particular when technical documentation is the object of study. The approach taken by QA systems is to allow a user to ask a query (formulated in Natural Language) and have the system search a background collection of documents in order to locate an answer.

In this paper we present a system developed to perform Question Answering[1] in technical domains. After initial experiences in the restricted context of the Unix man pages (Mollá *et al.*, 2000a; Mollá *et al.*, 2000b), we targeted the Aircraft Maintenance Manual (AMM) of the Airbus

[1]Here we deliberately use 'Question Answering' as a synonym for 'Answer Extraction' (Abney *et al.*, 2000), although we consider the latter a more fitting term to describe our work and most of the work currently done in this research sector.

A320 (Rinaldi *et al.*, 2002b; Rinaldi *et al.*, 2002c) and more recently we have embarked upon a new experiment, using the Linux HOWTOs as a new target domain.

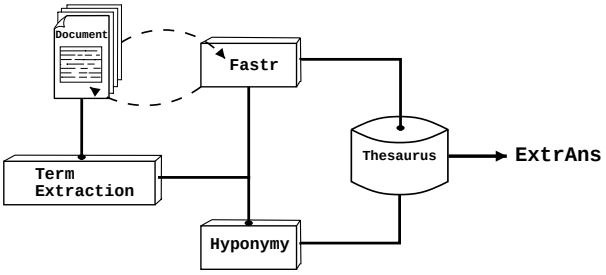


Figure 1: Term Processing

All these domains contain technical terminology that needs to be properly detected, managed and exploited before any NLP system can perform adequately. The AMM, which in source form is approximately 120MB large, describes how the constituent parts of an Airbus A320 relate to each other, the testing and maintenance procedures for each part, as well as the tools and materials to be used. As 30% of the words in the running text belong to the terminology, pre-processing needs to be focused in this direction. Terminology extraction followed by thesaurus construction are necessary first steps before using the terms in the Question Answering process (figure 1).

A desire for domain independent terminology extraction drives many "off-the-shelf" extraction tools (including commercial products) to target recall at the expense of precision. This bias may reduce their applicability to some specific tasks (Castellví *et al.*, 2001) but as "term candidates" need to be manually validated, over-extraction is preferable to missing terms. However, using the structure/nature of the analyzed text and designing simple extraction tools can be more effective in producing this initial list of candidate terms (Dowdall *et al.*, 2002).

The current focus, however, is the challenge of discovering relations implicit across these extracted terminologies. In particular, synonymy to conflate term variants into synsets, and hyponymy to create a taxonomy from these sets. The result of this phase of relation discovery is a domain thesaurus organized around synsets with each set representing a domain specific concept. This process could be described as an attempt to elicit hidden knowledge, implicit in the domain.[2]

Section 2 describes the operations adopted for structuring the terminology. Section 3 describes the use of this terminological thesaurus in our Question Answering system. Section 4 explores some related work. We refrain from a detailed evaluation of the system or individual techniques adopted, as the main focus of this paper is a descriptive one, the interested reader can find different types of evaluation in our previous work (Rinaldi *et al.*, 2002a; Rinaldi *et al.*, 2002b; Rinaldi *et al.*, 2002c). Further evaluations, in particular regarding the improvements described in this paper, are planned.

2. Structuring the Terminology

Despite all efforts in standardization, it is often unavoidable that different editors use different (but related) surface forms to refer to the same domain concept. Besides, new technical developments will lead to the continuous creation of new terms. Even in consolidated sectors there are no absolutely reliable methods to enforce standardization across different editors. Consequently, when processing technical documents it is vital to recognize not only standardized terminology but also potential variations and possible new terms.

[2] A recent survey of the epistemological status of the meta-terms 'term' and 'concept' in Terminology Theory can be found in Kageura (2002).

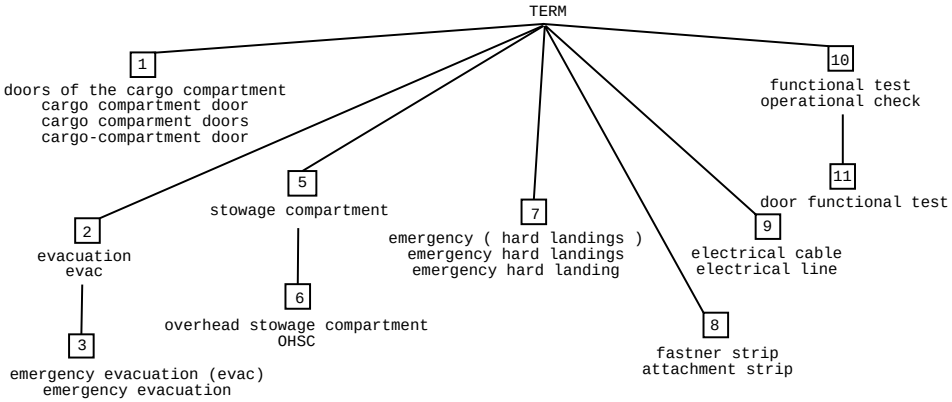


Figure 2: A sample of the AMM computational thesaurus

The process of terminological variation is well investigated (Ibekwe-SanJuan & Dubois, 2002; Daille *et al.*, 1996; Ibekwe-Sanjuan, 1998). The primary focus has been to use linguistically based variation to expand existing term sets through corpus investigation or to produce domain representations. However, a subset of such variations identifies terms which are strictly synonymous. The ExtrAns thesaurus gathers these morpho-syntactic variations into synsets. The sets are augmented with terms exhibiting three weaker synonymy relations described by Hamon & Nazarenko (2001). These synsets are organized into a hyponymy (isa) hierarchy, a small example of which can be seen in figure (2).

The first stage is to normalize any terms that contain punctuation by creating a punctuation free version and recording that the two are strictly synonymous. Further processing is involved in terms containing brackets to determine if the bracketed token is an acronym or simply optional. In the former case an acronym-free term is created and the acronym is stored as a synonym of the remaining tokens which contain it as a regular expression. So **evac** is synonymous with **evacuation** but **ohsc** is synonymous with **overhead stowage compartment**. In cases such as **emergency (hard landings)** the bracketed tokens can not be interpreted as an acronym and so are not removed.

The synonymy relations are identified using the terminology tool Fastr (Jacquemin, 2001). All tokens of each term are associated with their part-of-speech[3], their morphological root[4] and their synonyms[5]. How tokens combine to form multi-token terms is represented as a phrasal rule, the token specific information carried in feature-value pairs. Metarules license the relation between two terms by constraining their phrase structures in conjunction with the morphological and semantic information on the individual tokens.

Currently, we have designed the Metarules to identify strict synonymy that results from morpho-syntactic variation (**cargo compartment door** $\longrightarrow$ **doors of the cargo compartment**), terms with synonymous heads (**electrical cable** $\longrightarrow$ **electrical line**), terms with synonymous modifiers (**fastener strip** $\longrightarrow$ **attachment strip**) and both (**functional test** $\longrightarrow$ **operational check**). For a description of the frequency and range of types of variation present in the AMM

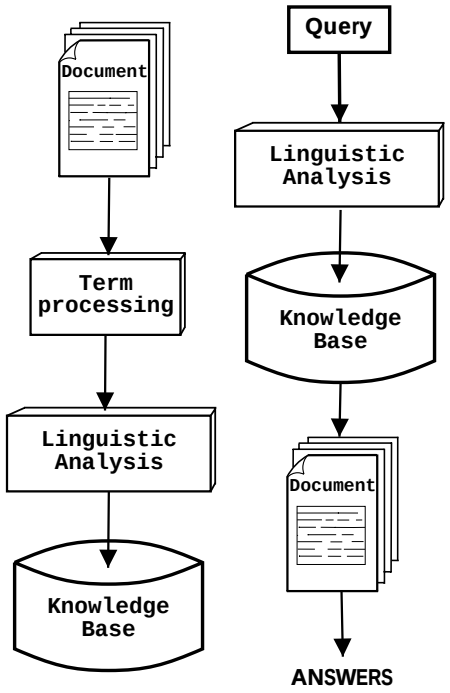
[3]assigned by the IMS TreeTagger, http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger
[4]obtained from CELEX, http://www.kun.nl/celex
[5]as defined by WordNet, http://www.cogsci.princeton.edu/wn

see Rinaldi *et al* (2002a).

A simple algorithm determines lexical hyponymy between terms. Term A is a hyponym of term B if: A has more tokens than B, all the tokens of B are present in A and both terms have the same head. There are three provisions. First, ignore terms with dashes and brackets as **cargo compartment** is not a hyponym of **cargo - compartment** and this relation (synonymy) is already known from the normalization process. Second, compare lemmatized versions of the terms to capture that **stowage compartment** is a hyperonym of **overhead stowage compartments**. Finally, the head of a term is the rightmost non-symbol token (i.e. a word) which can be determined from the part-of-speech tags. This hyponymy relation is comparable to the insertion variations defined by Daille *et al* (1996).

Automatically discovering these thesaurus relations across 6032 terms from the AMM produces 2770 synsets with 1176 hyponymy links. Through manual inspection of 500 synsets 1.2% were determined to contain an inappropriate term. A similar examination of 500 hyponymy links verified them all as valid.

3. Question Answering in Technical Domains



(a) Offline (b) Online

Figure 3: ExtrAns Processing Stages

Question Answering (**QA**, also called Answer Extraction) systems take a natural language query and return a small snippet of text which provides an answer from a predefined document collection. The field of Question Answering has flourished in recent years[6], in part, due to the QA track of the TREC competitions (Voorhees & Harman, 2001). These competitions evaluate systems over a common data set allowing developers to benchmark performance in relation to other competitors.

Extrans is a Question Answering system targeted at technical domains. The unix manpages provided a convenient testbed for experimentation with Answer Extraction techniques without the burden of a large document set (Mollá *et al.*, 2000b). Recent work on the Aircraft Maintenance Manual of the Airbus A320 (Rinaldi *et al.*, 2002b) has proved the scalability of the system to larger document sets and has offered a chance to solve problems related to SGML/XML formatting. Now we are moving back to the IT domain, considering in particular the Linux HOWTOs as a new document collection.

Processing is split into two distinct phases (figure 3). As we have seen, the first offline step is Term Processing involving extraction and organization of the term thesaurus. The next step, Linguistic Analysis, results in a semantic representation of the sentences – their **M**inimal **L**ogical **F**orm (see figure 4). These are stored along with their original location in a Knowledge Base. Online, the

[6] Although early work in AI already touched upon the topic, e.g. (Woods, 1977).

MLFs are a set of predicates in conjunctive form, where the variables are existentially bound. The main predications involve events, properties and objects, so multi-word terms are treated as standard objects. For example the MLF of (fig.5) is:

holds([1]),
object(electrical_coax_cable, o2, [v3]),
object(external_antenna, o3, [v4]),
object(ANT_connection, o4, [v5]),
evt(connect, [1] , [v3, v4]),
prop(to, p1, [[1] , v5]).

ExtrAns identifies three multi-word terms, translated as the objects: v3, a electrical_coax_cable, v4 an external_antenna and v5 an ANT_connection. The entity [1] represents the 'connect' event involving two arguments, the *electrical_coax_cable* and the *external_antenna*. This reified argument, [1] , is used again in the final clause to assert the event happens '*to*' v5 (the *ANT_connection*).

Figure 4: Minimal Logical Forms

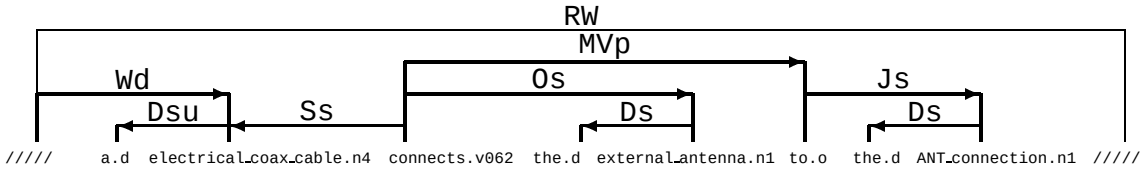


Figure 5: An example of LG output

user query is processed using the same linguistic analysis, and the resulting MLF is matched against the Knowledge Base. The matches are then displayed in the document so users can contextualize these potential answers.

Part of the Linguistic Analysis involves the Link Grammar parser (Sleator & Temperley, 1993), generating a dependency structure for each syntactic interpretation of a single sentence (figure 5). The multi-word terms from the thesaurus are identified and passed to the parser as single tokens. This prevents (futile) analysis of the internal structure of terms simplifying parsing by up to 50%.[7] This results in an average of 4.1 logical forms per sentence.

3.1. Extracting Answers

Answers are identified by matching (logically proving) the query MLF against the MLFs stored in the Knowledge Base. During construction of the MLFs, thesaurus terms are replaced by their synset identifier. This results in an implicit 'terminological normalization' for the domain. The benefit to the QA process is an assurance that a query and answer need not involve exactly the same surface realization of a term. Utilizing the synsets in the semantic representation means that when the query includes a term, ExtrAns returns sentences that logically answer the query, involving any of the terms' synset members.

[7] The measure refers to the average number of parses per sentence. As the removed parses are those which are incompatible with the (manually verified) terminology, we can be confident that this approach does not rule out potentially correct parses.

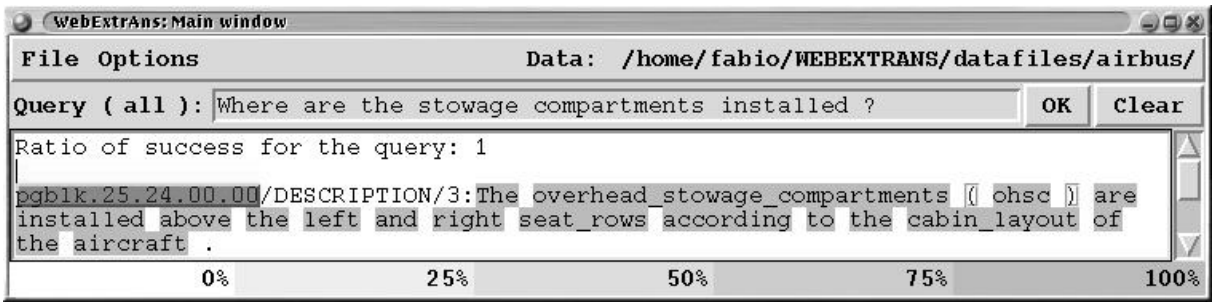


Figure 6: overhead_stowage_compartment is an hyponym of stowage_compartment

For example, the logical form of the query *Where are the stowage compartments installed?* is translated internally into the Horn query (1).

(1)
```
[evt(install,A,[B,C]),
object(D,E,[B]),
object(s_stowage_compartment,G,[C])]
```

This means that a term (belonging to the same synset as "stowage_compartment") is involved in an install event with an anonymous object. If there is an MLF from the document that can match example (1), then it is selected as a candidate answer and the sentence it originates from is shown to the user.

When the thesaurus definition of terminological synonymy fails to locate an answer from the document collection, ExtrAns taps the thesaurus hyponymy relations. Instead of looking for synset members, the Horn query is reformulated to included hyponyms and hyperonyms of the terms:

(2)
```
(object(s_stowage_compartment,A,[B]);
object(s_overhead_stowage_compartment,A,[B])),
evt(install,C,[D,B]),
object(E,F,[D|G])
```

Now the alternative objects are in a logical `OR` relation. This Horn query finds the answer in figure 6.

The expressivity of the MLF can further be expanded through the use of meaning postulates of the type: "*If x* **is installed** *in y, then x* **is** *in y*". This ensures that the query *"Where are the equipment and furnishings ?"*, extracts the answer *"The equipment and furnishings are installed in the cockpit"*.

A potential drawback of this search strategy is the strong reliance on the synonymy identification procedure. Should that fail, we might end up with 'ambiguous' synsets, containing terms that are not necessarily synonyms. Further, it is the very notion of synonymy that could be put into question, as it might happen that variants obtained with a series of transformational steps end up being very distant from the term that the process started from. It could be argued

that detection of synonymy cannot be always stated in boolean terms, i.e. in some cases we could say that two words or terms are synonyms to some degree.[8] However this problem, which is widespread in general language, is less relevant in technical domains, where the intended referent of a term is (in general) very precisely identifiable.

The level of ambiguous synsets in the AMM thesaurus (1.2%) is acceptable for ExtrAns' precision requirements. However, the definitions of the semantic relations between terms (especially synonymy) need to be tested and refined across different terminology intensive domains.

3.2. Present and Future Developments

The new domain of focus, the Linux HOWTOs and mini-HOWTOs, has the same pressing terminological needs as the AMM. The documents discuss, in practical terms, a variety of specific Linux topics (e.g. how to install the GNU C compiler or connect to a USB digital camera).[9] These documents will test the domain independence of the thesaurus construction techniques. Currently there are about 300 HOWTOs (with a total of approx. 3.5 million words) and about 150 mini-HOWTOs. Additionally, a subset are translated into languages other than English which will allow future work targeting cross-linguistic Question Answering.

The HOWTOs have been written in two similar markup languages: LinuxDoc and DocBook[10], for which both the SGML and XML versions of the DTDs are available. DocBook is a markup language created to support writing books and papers about computer hardware and software. The XML/SGML source of the HOWTOs mainly serves as a starting point in converting the HOWTOs into several different presentation formats (e.g. PDF, HTML, PostScript, etc.) and it is not intended as a support for 'intelligent' search through the documents (in particular terminology is marked up only partially, in an unsystematic fashion).

The SGML/XML formatting allows us to 'filter' with simple XML tools the zones of the documents that we intend to analyze (as in the case of the Airbus documents), easily excluding parts that are not suitable for this type of processing (figures, tables, etc.). Besides, customized visualization becomes easier using browsers of the latest generation. As we are particularly interested in terminology, the detailed sentence level markup that the DocBook language supports will be helpful for terminology extraction purposes, and the domain experts who could validate the extracted terminology could be easier to find in the Linux community.

At present, we have started tuning our linguistic processing and collecting domain-specific terminology from a selected subset of HOWTO documents (in English, covering approx. 600,000 words).[11] However the basic functionalities of the system have been proved very easy to port in this new domain, as can be seen in figure 7.

A limitation of the current version of our system is that only variants previously identified in the offline stage can be spotted in the user query. However it is always possible that the user comes up with a new variant, not previously seen, of an existing term. Although in theory it would be possible to generate all possible variants of the existing terms, that would be impractical because it would lead to very large synsets, and most of the variants will never be used.

We have developed and are currently testing a set of Metarules for Fastr targeted at this

[8] Unfortunately not easily measurable.

[9] Development of HOWTOs is part of the tasks of Linux Documentation Project, `http://www.tldp.org/`

[10] `http://www.oasis-open.org/docbook/`

[11] After removal of unanalyzable XML-zones (program samples, tables etc) and of markup tags.

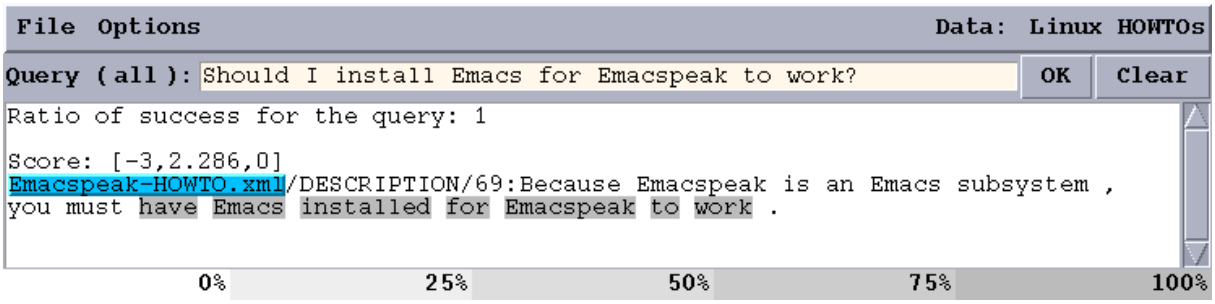


Figure 7: An example from the HOWTOs domain

problem. By filtering queries for these specific term variations, the need for a query to contain a "known" term from the thesaurus is removed. For example, the subject of the query *Where is the equipment for generating electricity?* is related through synonymy to the synset of **electrical generation equipment**, providing the vital link into the thesaurus.

The method of automatic thesaurus construction can map any semantic relation between words onto a term set. So far we have utilized only lexical hyponymy, however organizing the terms according to WordNet's (logical) hyponymy relations reveals new potential relations, like **surface protection** as a hyperonym of **floor covering**.

Another extension currently being developed is a new web-based interface, which will allow multiple users to query a centralized document collection with all of the ExtrAns functionality. The initial target domain will be that of the Linux HOWTOs.

4. Related Work

Within the medical domain, the Unified Medical Language System (UMLS), created by the National Library of Medicine[12] collects terminologies from differing sub-domains in a metathesaurus of concepts. The organization of the terms involve hyponymy and lexical synonymy. An application of the UMLS resource is PubMed[13] which retrieves the abstracts from medical journals by relating metathesaurus concepts against a controlled vocabulary used to index the abstracts. Comparing one controlled vocabulary against another elevates the term banks to a primary position in a kind of terminology based IR. This requires a complex, predefined semantic network of primitive types and their relations, but utilizing the terminology in this way makes the domain relatively accessible.

Cimino (2001) criticizes the UMLS because of the inconsistencies and subjective bias imposed on the relations by manually discovering such links. The alternative of a knowledge base of terminology (also in the medical domain) is explored, where terms are related by formal relations. The advantage of such approach is in the automatic methods which greatly facilitate thesaurus expansion.

Many Information Extraction (IE) tasks over this domain utilize the UMLS terminology in conjunction with shallow parsing in the construction of knowledge bases. A statistical *bag-of-*

[12] http://www.nlm.nih.gov/research/umls/
[13] http://www.ncbi.nlm.nih.gov/pubmed

words approach applied at the sentence level (Craven & Kumlien, 1999) determines predicate relations between proteins and chemicals, as long as multi-word terms are identified in the *bag*. Syntactically identifying object-predicate-object relations (Sekimizu *et al.*, 1998; Rindflesch *et al.*, 2000) would be impossible without the prior identification of multi-word term objects in the Metathesaurus. Inferences have also been directly extracted from the occurrence of terminology under certain of the MeSH headings (Cimino & Barnet, 1993). A term *X* under the abstract heading *methods*, and term *Y* under *diagnosis* implies that *X diagnoses Y*.

Hamon & Nazarenko (2001) explores the terminological needs of consulting systems. This type of IR guides the user in query/keyword expansion or proposes various levels of access into the document base on the original query. A method of generating three types of synonymy relations is investigated using general language and domain specific dictionaries.

5. Conclusion

In this paper we have described the crucial role of a terminological knowledge base in an AI system for Question Answering. While terminology extraction has been explored in many previous works, the importance of discovering relations among terms has often been neglected. In this paper we have presented our approach to the problem and showed the advantages that these techniques bring to a Question Answering system.

Références

ABNEY S., COLLINS M. & SINGHAL A. (2000). Answer extraction. In S. NIRENBURG, Ed., *Proc. 6th Applied Natural Language Processing Conference*, p. 296–301, Seattle, WA: Morgan Kaufmann.

CASTELLVÍ M. T. C., BAGOT R. E. & PALATRESI J. V. (2001). Automatic term detection: A review of current systems. In D. BOURIGAULT, C. JACQUEMIN & M.-C. L'HOMME, Eds., *Recent Advances in Computational Terminology*, p. 53–87. John Benjamins Publishing Company.

CIMINO J. & BARNET G. (1993). Automatic knowledge acquisition from medline. *Methods of Information in Medicine*, **32**(2), 120–130.

CIMINO J. J. (2001). Knowledge-based terminology management in medicine. In D. BOURIGAULT, C. JACQUEMIN & M.-C. L'HOMME, Eds., *Recent Advances in Computational Terminology*, p. 111–126. John Benjamins Publishing Company.

CRAVEN M. & KUMLIEN J. (1999). Constructing biological knowledge bases by extracting information from text source. In *the Seventh International Conference on Intelligent Systems for Molecular Biology, Heidelberg,Germany*, p. 77–86.

DAILLE B., HABERT B., JACQUEMIN C. & ROYAUT J. (1996). Empirical observation of term variations and principles for their description. *Terminology*, **3**(2), 197–258.

DOWDALL J., HESS M., KAHUSK N., KALJURAND K., KOIT M., RINALDI F. & VIDER K. (2002). Technical terminology as a critical resource. In *International Conference on Language Resources and Evaluations (LREC-2002), Las Palmas*. Available at `http://www.cl.unizh.ch/CLpublications.html`.

HAMON T. & NAZARENKO A. (2001). Detection of synonymy links between terms: Experiment and results. In D. BOURIGAULT, C. JACQUEMIN & M.-C. L'HOMME, Eds., *Recent Advances in Computational Terminology*, p. 185–208. John Benjamins Publishing Company.

IBEKWE-SANJUAN F. (1998). Terminological Variation, a Means of Identifying Research Topics from Texts. In *Proceedings of COLING-ACL*, p. 571–577, Quebec,Canada.

IBEKWE-SANJUAN F. & DUBOIS C. (2002). Can Syntactic Variations Highlight Semantic Links Between Domain Topics? In *Proceedings of the 6th International Conference on Terminology and Knowledge Engineering (TKE02)*, p. 57–64, Nancy.

JACQUEMIN C. (2001). *Spotting and Discovering Terms through Natural Language Processing*. MIT Press.

KAGEURA K. (2002). *The Dynamics of Terminology, A descriptive theory of term formation and terminological growth*. Terminology and Lexicography Research and Practice. John Benjamins Publishing.

MOLLÁ D., SCHNEIDER G., SCHWITTER R. & HESS M. (2000a). Answer Extraction using a Dependency Grammar in ExtrAns. *Traitement Automatique de Langues (T.A.L.), Special Issue on Dependency Grammar*, **41**(1), 127–156. Available at `http://www.cl.unizh.ch/CLpublications.html`.

MOLLÁ D., SCHWITTER R., HESS M. & FOURNIER R. (2000b). Extrans, an answer extraction system. *T.A.L. special issue on Information Retrieval oriented Natural Language Processing*. Available at `http://www.cl.unizh.ch/CLpublications.html`.

RINALDI F., DOWDALL J., HESS M., KALJURAND K., KOIT M., VIDER K. & KAHUSK N. (2002a). Terminology as Knowledge in Answer Extraction. In *Proceedings of the 6th International Conference on Terminology and Knowledge Engineering (TKE02)*, p. 107–113, Nancy. Available at `http://www.cl.unizh.ch/CLpublications.html`.

RINALDI F., DOWDALL J., HESS M., MOLLÁ D. & SCHWITTER R. (2002b). Towards Answer Extraction: an application to Technical Domains. In *ECAI2002, European Conference on Artificial Intelligence, Lyon*. Available at `http://www.cl.unizh.ch/CLpublications.html`.

RINALDI F., HESS M., MOLLÁ D., SCHWITTER R., DOWDALL J., SCHNEIDER G. & FOURNIER R. (2002c). Answer extraction in technical domains. In A. GELBUKH, Ed., *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, p. 360–369. Springer-Verlag. Available at `http://www.cl.unizh.ch/CLpublications.html`.

RINDFLESCH T., TANABE L., WEINSTEIN J. N. & HUNTER L. (2000). Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium on Biocomputing*.

SEKIMIZU T., PARK H. & TSUJII J. (1998). Identifying the interaction between genes and gene products based on frequently seen verbs in medline abstracts. *Genome Informatics*.

SLEATOR D. D. & TEMPERLEY D. (1993). Parsing English with a link grammar. In *Proc. Third International Workshop on Parsing Technologies*, p. 277–292.

E. M. VOORHEES & D. HARMAN, Eds. (2001). *Proceedings of the Ninth Text REtrieval Conference (TREC-9), Gaithersburg, Maryland, November 13-16, 2000*.

WOODS W. (1977). Lunar rocks in natural English: Explorations in Natural Language Question Answering. In A. ZAMPOLLI, Ed., *Linguistic Structures Processing*, volume 5 of *Fundamental Studies in Computer Science*, p. 521–569. North Holland.