Terminology as Knowledge in Answer Extraction

Fabio Rinaldi*, James Dowdall*, Michael Hess*,
Kaarel Kaljurand†, Mare Koit†, Kadri Vider†, Neeme Kahusk†

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

†Research Group of Computational Linguistics, University of Tartu
J. Liivi 2, 50409 Tartu, Estonia
{nkahusk, kaarel, koit, kvider}@psych.ut.ee

Abstract

It is well known that one of the greatest hurdles in automatically processing technical documentation is the large amount of specific terminology that characterizes these domains. Terminology poses two major challenges to the developers of NLP applications: how to identify domain specific terms in the documents and how to efficiently process them. In this paper we will present methodologies that we have used to extract and bootstrap a terminological database and its usage in an answer extraction system.

1. Introduction

In many domains the amount of existing and new terminology is so large that without adequate treatment it would soon become unmanageable. Consider for instance a large company, like an Aircraft Manufacturer with thousands of people working at different sites. If the terminology within the maintenance manuals is not extremely precise, the technicians may make mistakes, which could have very serious consequences.

Still, despite all efforts in standardization, 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.

In a recent project we aimed at processing the Aircraft Maintenance Manual (AMM) of the Airbus A320 within the context of an Answer Extraction (AE) system (Rinaldi et al., 2002a; Rinaldi et al., 2002b). This document (120MB) provided a new interesting domain for our previously developed Answer Extraction System (Mollá et al., 2000a; Mollá et al., 2000b), plus the additional challenge of a (mainly structural) SGML format. Many problems specific to this kind of technical domain fall within the area of terminology detection and management. Different materials, parts of the aircraft, technician's tools and units of measure are so abundant that without proper identification any NLP system would perform very poorly.

Existing terminology extraction tools have only limited reliability, therefore they can only serve as a starting point. We devised a strategy to collect domain-specific terminology from different external and internal sources and present it in an uniform repository. Three chapters of the AMM have been fully analyzed, all terms have been semi-automatically extracted and manually verified. The extracted terms served as a basis for an evaluation of various automatic term extraction tools and methods.

In this paper we will first present the details of the extraction process (section 2.). The extracted term list provided the input to the terminology tool FASTR (Jacquemin, 2001) to identify semantic relations (synonymy and hyponymy in particular) within the term set and the AMM (section 3.). In the final part of the paper (section 4.) we describe the usage of terminology in our Answer Extraction system, which originally motivated our involvement with terminology.

2. Terminology extraction

Different sources of information, both internal and external, were invaluable in the extraction process. First, several kinds of external sources (glossaries of abbreviations used in aircraft industry and different specifications, e.g.(ATA, 1997)) were used. Internally, different types of structures in AMM can indicate the presence of a term. Some of the terms are already explicitly denoted through the use of markup (e.g. element `CONNAME` for consumable material, element `TOOLNAME` for tools etc).

Existing terminology extraction tools have only limited reliability and they tend to be too general for some specific tasks (for different surveys of Terminology Extraction Tools see (Heidemann and Volk, 1999; Castellví et al., 2001)). Using some knowledge about the actual structure/nature of the analyzed text and rapidly designing simple terminology extraction tools can often be much more efficient. Two separate approaches have been considered and evaluated for extracting technical terms from the AMM manuals.

The first approach is based on a stop-phrase method that split certain SGML-zones (titles, paragraphs) using a list of phrases, units etc that often hint the presence of an adjacent term. For example, from a task title *Check of the Electrical Bonding of External Composite Panels with a CORAS Resistivity-Continuity Test Set* we cut out stop-phrases like emphof the, *of, with a* to obtain a list of candidate terms: Check, Electrical Bonding, External Composite Panels, CORAS Resistivity-Continuity Test Set. Given the high incidence of technical terms in the material we are

dealing with, even such a crude method can provide interesting results.

A second approach that we considered is a fully automatic statistical method by G. Dias et al ((Dias et al., 1999)). The method is very general, using no linguistic analysis, allowing n-grams to be of any length and allowing them to be non-contiguous (i.e. they can contain "holes"). It uses *Mutual Expectation* as an association measure, which evaluates the cohesiveness of a multi-word unit and a criteria called *LocalMax* to select the candidate terms from the evaluated list.

In order to simplify the manual verification and correction (either pruning or supplementing) of the extracted terminology, specific visualization tools have been developed. Early on in the project it was decided to convert the original SGML format of the manual into XML[1]. Using standard off-the-shelf tools we developed a simple XML-to-HTML converter that allows us to inspect the manual using a conventional browser. It is extremely helpful to be able to visualize the extracted terms in the context where they appear. In order to achieve this, additional XML markup that denotes the extracted units is inserted into the manual.

The new markup tags can be tied to presentational information (given e.g. by CSS stylesheets), so that when the manual is browsed the terms are highlighted and differentiated from the rest of the text. Most modern web browsers are capable of handling such specification of the information.

With a high degree of manual validation, aided in no small part by the visualization tools, the resulting term list is relatively complete. Against this list the automatic methods of terminology extraction can be evaluated.

The list obtained by the statistical method of Mutual Expectation and LocalMax (combined with simple stop-word filtering) showed the results of recall 44% and precision 15%.

For the list obtained by the stop-phrase method the recall was 66% and precision 12%. Better results can be explained by the fact that the stop-phrase method is aware of the structure of the manual and "knows" how important information and terminology is presented there, while the statistical method is general and makes only use of the frequencies and cohesiveness of the multi-word units.

When combining the methods, the recall grew to 78% and precision became 10%. Both the methods produced term-lists with relatively small intersection (only approx. 2000 terms).

We conclude that these methods are quite useful for obtaining a preliminary list of terminology, which can then be visualized to help manual checking. Even the statistical method which showed a low result of recall is still valuable for backing up the stop-phrase method.

3. Properties of the Terminology

The current section explores the properties exhibited by the extracted terminology and expands on the work previosly presented in (Dowdall et al., 2002). After discussing the distribution of terminology within the manual, we explore how Fastr (Jacquemin, 2001) uses syntactic, morphological and semantic information to distinguish synonymous terms. Finally we explain how the construction of hyponymy and meronomy hierarchies can disambiguate the internal syntactic structure of terms. These two processes are used to build and organize the terminological DB used in ExtrAns (see figure 1).

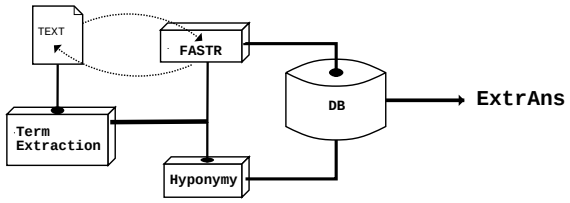

Figure 1: Use of Terminology in ExtrAns

3.1. Frequency distribution of the terminology

The main test-set for our terminological experiments was a combined list of terms from 3 chapters of the manual, containing approx. 1 million words[2].

Besides evaluating several automatic terminology extraction methods to find the most suitable for the AMM, we also analyzed whether the obtained results are general enough to be considered characteristic of the whole manual.

The list of terms extracted from the selected chapters contained approx. 13,000 terms of which approx. 1000 were single-word terms.

As we tried to extract all the possible spelling and morphological variants of the terms, we also grouped the terms so that each group contains all the spelling variants of one concept. This kind of grouping reduced the size of the list by 20%.

Unfortunately, the chapters share very few terms with each other, only about 250 terms are present in all three chapters, about 550 are present in two, the rest appear in only one chapter.

For one of the chapters 58% of the terms were only present there, 24% of the terms were shared by 2 to 4 chapters, and only 18% were a bit more general. The terms tend to be chapter-specific — the statistics show that each chapter is likely to contain its own unique terminology, which means that no chapter can be ignored in the process on terminology extraction.

For a great number of the terms the frequency of appearing in the manual is equal to one, which means that detecting them by frequency based methods is likely to fail.

According to our results, most of the terms are multi-word units, mainly bigrams and trigrams, but in principle there is no limit to the number of tokens a term can contain. Long terms usually denote material names or placards/messages, e.g.

[1]With some loss of information, though not relevant for our application.

[2]Those chapters are also the most frequently used by the technicians.

(1) USA MIL-S-81733 CLASS C CORROSION INHIBITIVE INTERFAY SEALANT

(2) system status message 'SLIDES PRESS LOW'

but also concepts like:

(3) bleed pressure regulator valve control solenoid

(4) flight crew electrical foot warmers

(5) hydraulically operated cargo compartment door

Still, most of the terms have a length of two or three tokens[3]. Bigrams and trigrams account for approx. 80% of the total amount of multi-word terms.

3.2. The role of FASTR

It is common to assume that terms are frozen compounds without variation (Sager, 1990), however in many practical cases this hypothesis proves to be remote from the actual reality of technical documentation. Often different notations for the same technical concept are introduced by different authors (including spelling variants, different word-order, use of synonyms, etc.). Most of these variations (Daille et al., 1996) are of a regular nature and can easily be predicted.

Fastr (Jacquemin, 2001) identifies linguistic variations on a base set of terms appearing in a text. The individual words involved in a previously extracted base set are associated with their part-of-speech [4], their morphological root[5] and their semantic synset[6]. Multi-word terms are represented as a feature structure of this information and Metarules licence variation from a base term to an occurance in the text. Designed as a terminology extraction tool, this process involves expanding a set of terms through corpus investigation. The degree of linguistic consideration involved in extracting new terms allows for an investigation of the types of relationships between the base set and the extracted variations.

The most simplistic variations are syntactic involving either inserting an argument (word or acronymn) into an existing term (6), permutating an existing syntactic structure (7) or coordinating existing terms (8).

(6) galley electrical system $\longrightarrow$
galley power electrical supply system

(7) water flow $\longrightarrow$ flow of water

(8) maximum rearward position $\longrightarrow$
maximum forward and rearward positions

This sort of variation is relatively productive accounting for 33% of the indexed variations. However, these simple syntactic variations are also involved in conjunction with types of morphological and semantic variation.

Purely morphological variations exchange morphologically related words (9). This type of substitution combined with a syntactic insertion (10) is rare (only two occurances), more common is a combination with a permutation on the original syntactic structure (11). Morphologically centered variations make up 25% of the indexed variants.

(9) electric connector $\longrightarrow$ electrical connectors

(10) electrical equipment $\longrightarrow$
electrically operated equipment

(11) differential pressure $\longrightarrow$ pressure difference

30% of the variants are semantic in nature. As with the morphology, a simple variation substitutes words. For a semantic variation the words must belong to the same synset, either heads or modifiers (12). Semantic substitution can be combined with insertion to define the relation between (13). The variant in (14) is a permutation of *cargo door* and is related to the base term as *load* and *cargo* belong to the same synset.

(12) bulk cargo $\longrightarrow$ bulk load

(13) minimum distance $\longrightarrow$
minimum handling space

(14) load door $\longrightarrow$ door for the cargo

The remaining variations (12%) were in punctuation and orthography:

(15) overhead stowage compartment $\longrightarrow$
overhead-stowage-compartment $\longrightarrow$
overhead stowage-compartment $\longrightarrow$
overhead stowage compartment(s)

(16) air grill $\longrightarrow$ air grille

These arbitrary differences represent strictly synonymous terms whereas (14) is the weakest useful synonymy relation, and there are as many differing degrees between these two extremes as one cares to discover. Where some studies (Hamon and Nazarenko, 2001) focus on these degrees, we take a conflation approach to these relations. As such, all variants of a single concept are grouped in WordNet type synsets.

3.3. Hyponomy

All complex nominal phrases carry structural ambiguity, unresolvable from consideration of the term alone. Is a 'toy car crusher' a 'toy' or does it crush 'toy cars' ? This question, sometimes referred to as NP bracketing (Barker and Szpakowicz, 1998), is concerned with exactly what is modifying what.

Intuitively, an *adjustable access platform* is an *access platform* which is *adjustable* (17). However, a *crew member seat* is a *seat* for a *crew member* (18) and an *underfuselage off-centered door* is a *door* that is both *underfuselage* and *off-centered* (19). Determining this information

[3]Here we mean by token a string of characters bordered by either a space or hyphen

[4]assigned by the IMS TreeTagger, see http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[5]obtained from CELEX, see http://www.kun.nl/celex

[6]as defined by WordNet, see http://www.cogsci.princeton.edu/wn

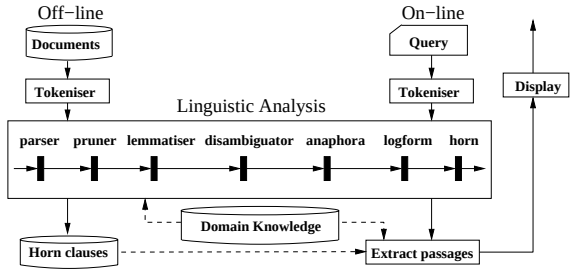

Figure 2: Architecture of the ExtrAns system

computationally is possible through investigating the hyponomy and mernonmy relations across the entire terminology.

(17) [adjustable [access platform]]

(18) [[crew member] seat]

(19) [underfuselage [off-centered [door]]]

For example, the existence of the hypernym parent of (17), *access platform*, as a term suggest the more specific term is created by simply modifying the parent to be *adjustable*. The lack of such a parent of (18), **member seat*, but the existence of the meronym term *crew member* suggests the phrasal modification of the head *seat*. The lack of any hypernym or meronym terms in (19) results in the individual modification of the single head.

Disambiguating terms in this way involves decomposing a term into all of its possible composite terms. For the three word term `[A,B,C]` this is `[A,B]`, `[B,C]` and `[A,C]` as inverted composites such as `[C,B]` or `[B,A]` are unrelated to the orignal term. The internal structure of the term is then a function of which of the valid composites actually appear as part of the terminology.

This type of investigation has the added advantage of producing automatic hyponomy and meronymy hierarchies, two useful lexical networks. As we have seen, it is conceptually possible to segment a multi-word term in a number of ways but only some divisions are coded in the vocabulary of the terminology set. These divisions help disambiguate the internal structure of the terms.

4. ExtrAns

An Answer Extraction (AE) system aims at extracting from given documents explicit answers to arbitrarily phrased questions. Over the past few years our research group has been working on the ExtrAns system which so far has two domains of application. Originally the domain of Unix manpages was chosen as a convenient testbed. Later we considered the Aircraft domain and started to work on the Mantainance Manual of the Airbus A320.

Extrans processes sentences and produces a core semantic representation which facilitates the semantic comparison of queries against text (see figure 2). Fundamentally this proccess can be divided into three stages, syntactic analysis, disambiguation and semantic generation.

The syntactic analysis begins with the tokenizer. Sentences are split into the units of analysis which optimize processing - words, sentence boundaries and terminolgy are all identified. As the head of a multi-word term controls sentence level syntactic behaviour, each term is considered as a single unit and assigned the syntactic requirements of the head. As such, all terminology is identified as either `TERM.p` or `TERM.s`. In this way terms are parsed as either singular or plural nouns.

Parsing involves the robust, dependency-based formalism of Link Grammar (LG)(Sleator and Temperley, 1993). Each word carries linking requirements (singular determiners 'look for' singular nouns etc.), a linkage representation of a sentence (fig.3a) satisfies all of these individual requirements in a connected graph without any cross-over links. Processing multi-word terms as individual tokens would introduce additional linking requirements. In the best case, modifiers are all connected to the head (fig.3b), identifying the term as a phrasal unit but offering only a superficial representation of the internal structure. In more complex sentences, such modifiers can often also link to words outside the term, resulting in multiple parses for the given sentence. The single token approach resulting in (fig.3a) stops such ambiguities from ever arising and saves the computational expense needed to disambiguate between the alternatives.

It is remarkable that usage of pre-detected terminology can simplify the parsing process (in terms of time and space) by as much as 50%. This is probably due to the highly technical nature of our domain, with a high incidence of domain specific terminology which could not be processed efficiently by any standard parser.

Such expense would be worthwhile if an accurate internal representation of the terminology were possible. As LG is not a noun phrase grammar, all terms are assigned the structure in (fig.3b), additional modifiers add `A` (adjectival modifier) or `AN` (nominal modifier) links to the head of the phrase. Whilst this structure may correctly describe some terms (*underfuselage off-centered door*), arbitrary application to *air conditioning system*, *electrical coax cable* or the extension to *no smoking/fasten seat belt (ns/fsb) signs* fails to capture the more subtle patterns of modification.

The directed dependency relations are used to express verb-argument relations, as well as modifier and adjunct relations. This expression becomes the Minimal Logical Form (MLF) that encodes the fundamental meaning of sentences. The MLFs represent a powerful combination of selected reification and underspecification.

The MLF are expressed as conjunctions of predicates with all of the variables existentially bound with wide scope. The main predications involve events, properties and objects, so multi-word terms are treated as standard objects. For example the MLF of (fig.3a) is:

(20) 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]).

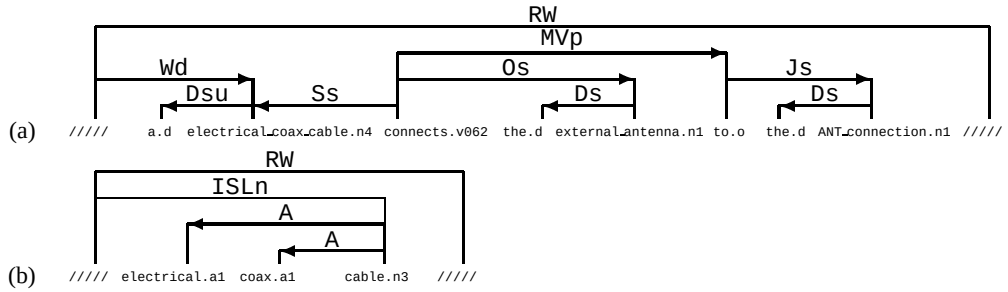

Figure 3: An example of LGs output

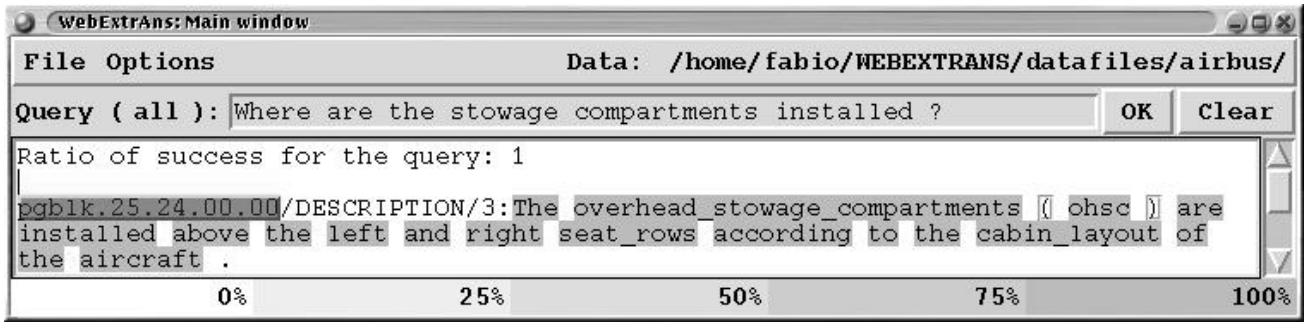

Figure 4: overhead_stowage_compartment is an hyponym of stowage_compartment

ExtrAns identifies three multi-word terms, translated into (20) 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*).

ExtrAns finds the answers to questions by processing queries in the same manner and matching their MLFs against the stored MLFs of the documents.

4.1. Variants as Synsets

Terms are stored in a DB in a way that the same term will receive a different synctactic identifier according to whether it was singular or plural (TERMs, TERMp) and a different semantic identifier which is the number of the synset to which it belongs.

In this way the same term (or terms belonging to the same synset) are treated syntactically as either singular or plural noun phrases, however semantically they are considered identical[7].

A possible alternative approach would be to exploit the internal structure of terms, as explored in 3.3.. This would however require mantaining a dual representation for each term at various levels of processing, once a as a frozen syntactic unit (useful for parsing) and once as a compound, where the head carries the syntactic information. At present, we find such an approach to be cumbersome while the solution that we have adopted provides for a neater flow of information. We do not rule out however the possibility of exploting the internal structure of terms at a later stage in our research.

Variations such as those presented in section 3.2. represent differing degrees of synonymy. From strictly synonymous spelling or punctuation variations to weakly related terms (see example 14), with many intermediate classifications. As such, all the detected variations of the same technical concept are collected within a WordNet type synset.

While processing the manual, all the variants are replaced by their synset number. The query is then processed in the same fashion. This approach leads to a degree of normalization for what concerns terminology representation. In the application this removes the need for a query term and a document term to be identical, variations in terminology between query and document no longer prohibits the logical proof criteria.

4.2. Extended Search

In the extended search mode the logical form of the query is extended taking into account hyponyms and hyperonyms of the tokens or terms that it contains. The logical form (21) of the query *Where are the stowage compartments installed?* is translated into the Horn query (22).

(21) holds(v5)**,**
evt(install, v5, [v2, v4])**,**
object(stowage_compartment, v1, [v4])**,**
object(anonym_object, v3, [v2])**.**

(22) [evt(install,A,[B,C])**,**
object(D,E,[B])**,**
object(s_stowage_compartment,G,[C])]

This means that an object which is a stowage_compartment is involved in a install event

[7] Generally speaking, a term does not necessarily have to be a noun phrase, though this is true in our domain.

with an anonymous object. If there is a MLF from the document that can identify the anonymous object (i.e. where the install event is) the answer is found. If not an expansion of the Horn query to included all hyponymy and hyperonymy possibilities is tried:

(23) (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 4.

5. Conclusion

In this paper we have described some problems related with terminology identification in a specific, highly technical, domain. We have presented some tools and techniques that have eased to some our degree our work. Finally we have discussed the usage of the terminology within an Answer Extraction system. We are confident that our experience can be relevant for a large number of technical applications.

6. References

ATA, 1997. *ATA Common Support Data Dictionary for The Air Transport Industries*.

K. Barker and S. Szpakowicz. 1998. Semi-automatic recognition of noun modifier relationships. In *Proc. of the 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics (COLING/ACL-98)*, pages 96–102.

M. Teresa Cabré Castellví, Rosa Estopà Bagot, and Jordi Vivaldi Palatresi. 2001. Automatic term detection: A review of current systems. In Didier Bourigault, Christian Jacquemin, and Marie-Claude L'Homme, editors, *Recent Advances in Computational Terminology*, pages 53–87. John Benjamins Publishing Company.

B. Daille, B. Habert, C. Jacquemin, and J. Royaut. 1996. Empirical observation of term variations and principles for their description. *Terminology*, 3(2):197–258.

Gaël Dias, Sylvie Guilloré, and José Gabriel Pereira Lopes. 1999. Multilingual aspects of multiword lexical units. In *Proceedings of the Workshop Language Technologies — Multilingual Aspects*, pages 11–21, Ljubljana.

James Dowdall, Michael Hess, Neeme Kahusk, Kaarel Kaljurand, Mare Koit, Fabio Rinaldi, and Kadri Vider. 2002. Technical terminology as a critical resource. In *International Conference on Language Resources and Evaluations (LREC-2002), Las Palmas*, 29–31 May.

Thierry Hamon and Adeline Nazarenko. 2001. Detection of synonymy links between terms: Experiment and results. In Didier Bourigault, Christian Jacquemin, and Marie-Claude L'Homme, editors, *Recent Advances in Computational Terminology*, pages 185–208. John Benjamins Publishing Company.

Britta Heidemann and Martin Volk. 1999. Evaluation of terminology extraction tools: TExt for TWIN, System Quirk, Xerox TermFinder. Technical report, University of Zurich.

Christian Jacquemin. 2001. *Spotting and Discovering Terms through Natural Language Processing*. MIT Press.

Diego Mollá, Gerold Schneider, Rolf Schwitter, and Michael Hess. 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.

Diego Mollá, Rolf Schwitter, Michael Hess, and Rachel Fournier. 2000b. Extrans, an answer extraction system. *T.A.L. special issue on Information Retrieval oriented Natural Language Processing*.

Fabio Rinaldi, James Dowdall, Michael Hess, Diego Mollá, and Rolf Schwitter. 2002a. Towards Answer Extraction: an application to Technical Domains. In *ECAI2002, European Conference on Artificial Intelligence, Lyon*, 21–26 July.

Fabio Rinaldi, Michael Hess, Diego Mollá, Rolf Schwitter, James Dowdall, Gerold Schneider, and Rachel Fournier. 2002b. Answer extraction in technical domains. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 360–369. Springer-Verlag.

Juan C. Sager. 1990. *A Practical Course in Terminology Processing*. John Benjamins, Amsterdam.

Daniel D. Sleator and Davy Temperley. 1993. Parsing English with a link grammar. In *Proc. Third International Workshop on Parsing Technologies*, pages 277–292.