Technical Terminology as a Critical Resource

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

[*]University of Zürich, Institute of Computational Linguistics
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

Technical documentation is riddled with domain specific terminology which needs to be detected and properly organized in order to be meaningfully used. In this paper we describe how we coped with the problem of terminology detection for a specific type of document and how the extracted terminology was used within the context of our Answer Extraction System.

1. Introduction

Terminology is know to be one of the main obstacles in analyzing text using NLP techniques. This is particulary true in the case of manuals, where a significant amount of the text is made up of technical terms.

In a recent project we have dealt with the domain of Aircraft Maintenance Manuals (AMM) (Rinaldi et al., 2002). We obtained a machine readable manual for the Airbus A320 and aimed at processing it within the scope of an Answer Extraction (AE) application. An AE system aims at extracting from given documents explicit answers to arbitrarily phrased questions.

The manual has a mainly structural SGML format and it is of considerable size (120MB). Many problems specific to this kind of technical domain (described in section 2.). 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 based on the EuroWordNet (EWN) format. Three chapters of the AMM have been fully analyzed, all terms have been semi-automatically extracted and then manually verified. The extracted terms served as a basis for an evaluation of various automatic term extraction tools and methods. Detailed results and comparisons are presented in section3.

Despite commonly held assumptions regarding uniformity of terminology (Sager, 1990), in many real cases different notations for the same technical concept are introduced (including spelling variants, different word-order, use of synonyms, etc.). These variations (Daille et al., 1996) are often of a regular nature and can easily be predicted. Some techniques to detect these variants are presented in section 4.

The extracted terminological DB has been used within our Answer Extraction system, as described in section 5.

One of the positive results is the significant reduction in the complexity of parsing. Our results show that pre-detected terminology can simplify the parsing process (in terms of time and space) by as much as 50%.

2. AMM's terminology

The Aircraft Maintenance Manual made available to us is divided into 46 chapters. We concentrated our attention on the three most often queried chapters which cover approx. 1 million words, making 10% of the full manual. Several automatic and semi-automatic methods were used for terminology extraction purposes. We ended up with term-lists of approx. 13000 terms and variants (including spelling and morphology). Considering this list to be relatively complete, in terms of the selected chapters, enabled us to explore different features of both the analyzed document and the terminology itself.

First we look at the markup used in AMM (section 2.1.), then the variation of the terms (section 2.2.), and finally several frequency distribution characteristics of the terms (section 2.3.). All those features are closely related to possible terminology extraction principles. Although we have experimented with a single type of document, our results are likely to be of relevance for many similar types of document, with a high occurence of terminology in a highly technical domain.

2.1. The role of AMM markup in defining terms

The manual was originally structured in SGML, but since it almost always followed the XML requirements (e.g start-tags must have corresponding end-tags, attribute values must be in quotation marks etc) it was easily converted into XML. Such conversion was mainly needed to be able to use the growing number of XML tools for processing the manual.

Technical documentation would benefit hugely if all its terminology were explicitly denoted through the use of markup tags, e.g. this would make a keyword-based information retrieval possible. Still, if this is not the case

and markup is used mainly for other purposes (e.g. to define layout) concentrating attention on specific markup elements can still be helpful when searching for terms.

The analyzed manual is basically a sequence of lists and tables, the maintenance tasks are mostly specified by "steps-to-be-taken" lists. The terminal node in the XML tree structure is in most of the cases PARA which denotes paragraphs. It is used inside list items and table cells (ENTRY) as well as separately. PARA can contain several elements (CONNAME, TOOLNAME etc) that are used to denote names of different entities e.g. tools, materials etc. Another important element is TITLE which specifies the heading of a chapter/section. PARA and TITLE cannot contain each other.

The following table illustrates the "term-consistency" of the mentioned zones in terms of recall[1] and precision[2].

XML zone	Recall	Precision
PARA	94.5%	18.9%
ENTRY	16.4%	25.6%
TITLE	10.1%	42.0%
*NAME	1.6%	60.1%

Note that the paragraphs which cover most of the text in the manual have surprisingly high precision, meaning that sentences in the manual are basically made out of terms and function words, and not much else.

2.2. Variation of Terms

A perhaps unexpected and to some extent surprising feature of the terminology used in the AMM is the relatively high amount of variants that can be found.

A term can often be spelled in several different ways, either containing hyphens instead of spaces, or even omitting the spaces. Sometimes uppercase characters are used (e.g. the term is part of a heading), sometimes only the first characters of each word is capitalized. Also terms can be in plural or the plural can be optional (in this case parentheses are used). Spelling variation as exemplified below appear very often:

(1) CARGO COMPARTMENT DOOR
    Cargo Compartment Door
    Cargo-Compartment Door
    cargo compartment door
    cargo compartment door(s)
    cargo compartment doors
    cargo-compartment door

More subtle variations (morphological, syntactical and semantic) appear with an alarming frequency, such that it would be impossible not to take them into consideration. In section 4. this problem is discussed at length.

---

[1]recall was calculated by checking how many terms from our final list were present in the zone

[2]precision was calculated by first deleting all the terms from the zones by simple search-replace method and then measuring the amount of garbage left behind. Low precision means lots of garbage
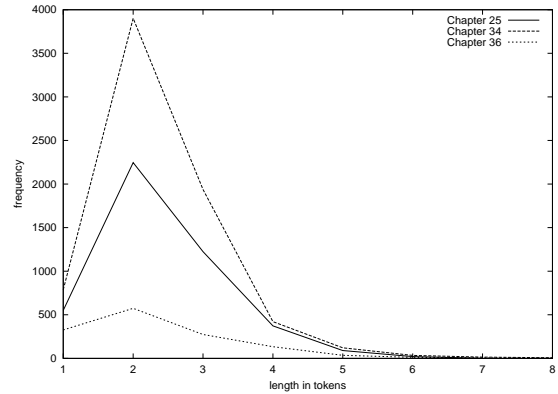

Figure 1: Frequency distribution of extracted terms from the selected chapters by the number of tokens they contained

2.3. Frequency distributions of terms

Another relevant propriety of the terminology in the AMM is its uneven distribution. In particular it is possible to show that it tends to be chapter-specific. In the next section it is explained how terminology for three selected chapters was extracted. Starting from those chapters, it is possible to make predictions about the rest of the manual and its terminology and calculate several statistical figures about the term-list itself.

As it turned out, the selected 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. Generally, very few of the extracted terms appear in the rest of the manual, meaning that terms tend to be chapter-specific. Therefore, no chapter should be ignored in the process of terminology extraction, meaning that it is likely to take a lot of time, when based on manual or semi-automatic methods.

For a great number of 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 multiword units, mainly bigrams and trigrams (which together make 80% of all terms), but in principle there is no limit to the number of tokens a term can contain (see figure 1). Long terms usually denote material names or placards/messages, e.g.
USA MIL-S-81733 CLASS C CORROSION INHIBITIVE INTERFAY SEALANT
but also concepts like:
bleed pressure regulator valve control solenoid

3. Terminology extraction and representation

Different sources of information, both internal and external, were invaluable in the extraction process. First, several kinds of external sources were used, like glossaries of abbreviations used in aircraft industry (ATA, 1997), different specifications, etc. More importantly, internal sources,

i.e. the manual itself must be processed in order to extract terms. This way we are able to obtain the terminological units that actually appear in the text together with the variety of forms that are presented there (see 2.2.).

Existing terminology extraction tools have only limited reliability and they tend to be too general for some specific tasks.[3] Using some knowledge about the actual structure/nature of the document and rapidly designing simple terminology extraction tools can often be much more efficient.

Different types of structures in AMM can indicate the presence of a term (see 2.1.). Selecting XML-zones that contain terms with relatively high precision, offers the possibility to use simple and yet reliable methods for terminology extraction. Even if the recall tends to be low, those terms can serve as a seed and reference for more advanced methods.

Also specific patterns that are used when talking about terms are quite frequent in AMM. For example, terms are often capitalized and followed by an acronym in parentheses as in *Power Transfer Unit (PTU)*. Or the other way around: *PTU (Power Transfer Unit)*. These patterns can be detected and properly processed using simple regular expressions. Terms are often followed by a reference/link to an other section of the manual. Again, the presence of such reference can be taken as an indication of an adjacent term.

This kind of document-specific approach turns out to be useful, as well as inevitable, since general terminology extraction tools are not aware of such structures. Still, the described methods work only on a limited number of structures and therefore more general methods have to be used.

3.1. Stop-phrase method

Our first approach was based on a stop-phrase method that split certain XML-zones (first titles, later paragraphs) using a semi-automatically compiled 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 *of the*, *of*, *with a* to get a list: 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.

3.2. Statistical method

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

This method was implemented as a Perl program and applied to the selected chapters of the manual. The markup tags were removed from the text beforehand.

Due to the low precision of the result, two stop-word filters were used to prune the output, they either removed the extracted units from the list or modified them by deleting words from the beginning and end. This kind of stop-word filtering reduced the length of the list by about 30%.

3.3. Visualization

In order to simplify the manual verification and correction (either pruning or supplementing) of the extracted terminology, we developed simple visualization tools for viewing the manual with a conventional web browser.

The XML formatting of the manual enabled us to use standard off-the-shelf tools for handling style specifications in XSLT and/or CSS. Additionally, new markup tags were inserted into the manual to denote the extracted units.[4]

Tying the new tags to style information makes it possible to see the extracted terms highlighted in the context, and to correctly decide whether the textual unit really functions as a term or not (see figure 2). This can be difficult without any kind of visualization since non-terms can often resemble real terms syntactically (e.g. 'wall-mounted cabin attendant' can be retrieved by the extraction software instead of 'wall-mounted cabin attendant seat'). Visualization is a simple and effective way of coping with the noise produced by many of the current terminology extraction tools and it is even more crucial when the viewer is not an expert in the domain.

3.4. Evaluation of automatic methods

After the manual checking of the terms we got a term-list that we considered relatively complete regarding the selected chapters. This gave us a chance to further test and evaluate the fully automatic methods for terminology extraction and to conclude how much manual work had to be done to refine the automatically extracted terminology.

As it turned out, 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% while the stop-phrase method produced lists of recall 66% and precision 12%.

When combining the methods, by taking a union of the lists, recall grew to 79% and precision became 10%. Combining by intersection recall shrank to 32% and precision became 63%.

Relatively low results of recall and precision leave a lot of manual work to be done afterwards. Of course, both methods can be improved, mainly by using several document-specific tricks, e.g. adding stop-words and using more intelligent preprocessing of the document. Then again, it might be more efficient to take the preliminary lists and continue the work manually in a convenient visualized environment.

---

[3]For a survey of Terminology Extraction Tools see for instance (Heidemann and Volk, 1999).

[4]This can be achieved by a simple substitution mechanism e.g. by using a standard Unix tool *sed* that searches for each extracted term in the text and borders it with given markup tags.

Job Set-up

- Energize the [[ground service] network] 24–42–00–861–001 .

- Access to the [Inflation Reservoir] (10)

  401/TASK 25–62–46–991–001–A
  - Open the AFT [[cargo–compartment] door] 52–30–00–860–001 .
  - Put an [access platform] in position adjacent to the AFT [[cargo–compartment] exit].
  - Set the AFT [CARGO COMPT [light switch]] 8LU to ON.
  - FOR **7506MM**
    - Remove the [access panel] 151KW .
  - FOR **7507MM**
    - Remove the [access panel] 152KW .

Figure 2: Viewing the manual in a web browser

3.5. Thesaurus Representation in EWN

The aims of a Thesaurus is to support (1) query analysis and (2) creation of the knowledge base to assure that the appropriate elements refer to one and the same thing or process in the reality. For example, a part of the aircraft can be referred to by a technical name, abbreviation or reference number. All these lexical units together build up a synonym set (synset) in a single entry.

The database structure of the Thesaurus is built according to EuroWordNet (Vossen, 1997; Vossen, 1998)) and was built in its initial form with Polaris tool (Louw, 1998).

The Thesaurus is centered around conceptual units, which are equivalent to WorNet's synset. Each synset is built from synonymous words or compound phrases. The synsets are connected with semantic relations, mostly hyperonym/hyponym relations, occasionally meronym relations as well.

```
entry number
   literal
      [definition, examples of usage]
      [information about the element
      (e.g. term) and the source]
   /---/
   literal
      /---/
   [semantic relations (to all literals
    in one entry)]
   [the label of the semantic relation
    and the first literal of the other
    entry]
```

To build the domain-specific Thesaurus, we included only the 'standard' forms of words and acronyms into synsets. The other variants were included into additional term lists and linked to corresponding synsets. At present there are 12 370 entries (synsets).

In the final version, the Thesaurus was converted from EWN import-export format into an XML format specially designed for this purpose. A suitable DTD was developed, which offers an abstract view of the thesaurus' structure and can be used to track down errors. The thesaurus is mainly centered around the hyperonym/hyponym relation, however other relations are considered. The new XML format offers a convenient and platform-independent way of exploring the thesaurus, via a conventional browser, with the help of simple visualization tools based on XSL and CSS stylesheets.

4. Lexically Relating Terms

4.1. Synonymy

Without enforced homogenicity terms are subject to the same variation as all noun phrases. Variation in how a single concept is represented in the terminology comes from two sources - arbitrary editorial differences introduced by different authors (section 2.2) and natural patterns of language variation.

For some of the extracted terminology there is a one to one mapping from term to concept. Terms such as *actuating mechanism* and *potable water pipe* are consistently used throughout the manual to refer to the entities they denote.

Other terms, however, represent only one way a concept is referred to in the manual. The 'alternative' references arise from natural patterns of noun phrase transformation. Either through syntactic permutation:

(2) cargo compartment door $\longrightarrow$ door of the cargo compartment

Or through the use of morphologically related tokens:

(3) cargo heat system $\longrightarrow$ cargo heating system

Or a combination, producing a morphosyntactic variant:

(4) cabin floor cover $\longrightarrow$ covering on the cabin floor

Such variations appearing in the manual can be automatically detected. 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 [5], their morphological root[6] and their semantic synset[7]. 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.

Using WordNet as one of the lexical resources additionally identifies the kinds of synonymy relations investigated by (Hamon and Nazarenko, 2001):

(5) vertical position $\longrightarrow$ upright position
     bulk load $\longrightarrow$ bulk cargo
     functional test $\longrightarrow$ operational check

4.2. Hyponomy and Meronomy

All methods of term extraction must address the problem of term verification. Our visualization tools (3.3.) allow terms to be contextually verified and hint at the lexical relations that a complex term has with other members of the terminology. Exploring such relations fully is a simple matter of decomposing a term into all of its possible composite terms, then verifying which of these possibilities are actually part of the terminology.

Considering a mulit-word term of three tokens [A,B,C] the possible two token terms are [A,B], [B,C] and [A,C]. So the term *adjustable access platform* can be decomposed into:

(6) adjustable access
     access platform
     adjustable platform

There is no need to consider composites that do not preserve the ordering of the original term such as [C,B] or [B,A] as, even if they exist, they would be unrelated to the original term.

Pattern matching across the entire terminology determines that the only valid composite in (6) is *access platform*, and as this preserves the head of the original term it is a hyperonym. Such hyperonym terms can also come from an [A,C] composite which is separated by additional token to form the three token original, *air extraction hose* contains the hypernym *air hose*. These two hyponomy possibilities are sometimes present in a single term, so *detailed visual inspection* contains the hyperonyms *visual inspection* and *detailed inspection*.

The final composite possibility for a three token term, [A,B], frequently appears across the terminology, but does not indicate a hyponomy relation. For example, *crew member seat*, is the result of adding a head to the existing term *crew member*. We define the relations between such terms as meronomy in reference to the part-whole relationship between the original and the composite which does not result in hyponomy.

This approach applies to terms of any length, with an increase of composite possibilities introduced by longer terms. So the four token term *galley power supply system* yeilds the hypernym terms (7) and the meronym terms (8).

(7) galley supply system
     power supply system
     supply system

(8) galley power supply
     galley power
     power supply

4.3. Terms as Synsets

Terms and their variants are stored into a data structure inspired by Wordnet. All the variants of a given term belong to the same synset, which is then given a unique numerical identifier. This identifier is used in later processing as a substitute for the term and the variants. Synsets that contain terms participating in an hyper-/hyponym relation are then correspondingly marked. In this way, exploiting all the identified term relations, it is possible to build a domain-specific mini-ontology, which can be useful for many purposes.

In order to speed up the identification and processing of the terms, they are split into two separate lists (plurals and singulars). From the existing singulars the corresponding plurals are generated and viceversa, removing duplicates if necessary. The two lists are then stored separately in a DB in a way that the same term will receive a different syntactic identifier according to whether it is singular or plural (TERMs, TERMp) and a different semantic identifier which is the number of the synset to which it belongs.

5.   The ExtrAns system

Over the past few years our research group has been working on an Answer Extraction System (ExtrAns, see (Mollá et al., 2000; Rinaldi et al., 2002)), which aims at extracting from given documents explicit answers to arbitrarily phrased questions. Initially, we chose the domain of Unix manpages as a convenient testbed. More recently, we moved to the Aircraft domain and started specific work on the Mantainance Manual of the Airbus A320.

These two domains represent small to medium sized document collections, which have the obvious advantage that it is still possible to process the entire document collection in an off-line stage, rather than just selected documents (or paragraphs) at run-time. As the data sets continue to grow in size this approach will quickly become too computationally expensive and paragraph indexing methodologies will need to be used. Currently, a simple pre-selection mechanism, based on a loose matching between query concepts and the stored semantic representation of the document, ensures that the search time remains within reasonable limits.

The ExtrAns system is based on a common syntactic and semantic processing of documents and queries. The documents are processed off-line and their semantic representation stored in a knowledge base, while the query is processed at run-time and its semantic representation matched against the knowledge base.

The syntactic processing is based on the Link Grammar (LG) parser (Sleator and Temperley, 1993), which is a robust and efficient dependency-based parser (see figure 4 for an example of a parsed sentence). Some of the

---

```
                                              RW
            ┌───────────────Wd────────────────┐┌──────────────Os──────────────┐
            │       ┌───Ds───┐    ┌────Ss────┐│     ┌──────Ds──────┐           │
///// the    <toggle_switch>    controls    the   <cargo_compartment_door>    /////
```
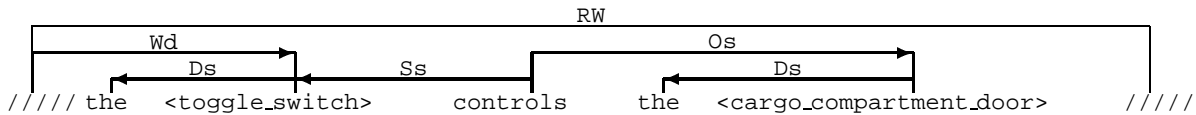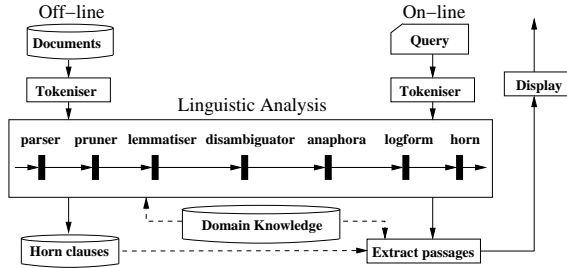
Figure 4: An example of LG output

Figure 3: Architecture of the ExtrAns system

returned parses are filtered out using a corpus based approach (Brill and Resnik, 1994) (disambiguating in particular prepositional phrase attachment or gerund and infinitive constructions). An anaphora resolution algorithm (Lappin and Leass, 1994) is used to find the referents of sentence-internal pronouns. A semantic representation (Minimal Logical Form - MLF) for each of the remaining parses is produced using the dependency relations generated by LG. The MLFs are expressed as conjunctions of predicates with all the variables existentially bound with wide scope.

Pointers to the original text attached to the retrieved logical forms allow the system to identify and highlight those words in the retrieved sentence that contribute most to that particular answer. An example of the output of ExtrAns can be seen in Fig. 5. When the user clicks on one of the answers provided, the corresponding document will be displayed with the relevant passages highlighted.

When no direct proof for the user query is found, the system is capable of relaxing the proof criteria in a stepwise manner. First, hyponyms of the query terms will be added, thus making it more general but still logically correct. If that fails, the system will attempt approximate matching, in which the sentence with the highest overlap of predicates with the query is retrieved. The (partially) matching sentences are scored and the best fits are returned. In the case that even this method does not find sufficient answers the system will attempt keyword matching, in which syntactic criteria are abandoned and only information about word classes is used. This last step corresponds approximately to a traditional passage-retrieval methodology with consideration of the POS tags. It is important to note that, in the strict mode, the system finds only logically correct proofs (within the limits of what MLFs can represent; see below), i.e. it is a high precision AE system.

5.1. Terminology in ExtrAns

Domain-specific terminology poses various problems to any parser. First there will be a significant amount of unknown words, which would have to be manually added to a lexicon. Second, the internal structure of the terms is often unconventional (Justeson and Katz, 1995) and the parser would have to spend additional effort in trying to detect possible internal structure as well as many clearly incorrect parses when the multi-word term is split and its words differently combined within the context of the sentence.

The ExtrAns tokenizer is capable of splitting the text into the units of analysis which optimize processing - words, sentence boundaries and terminolgy are all identified. Terms are considered as single units and assigned the syntactic requirements of their head word. 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.

At the same time, the terms will be assigned their corresponding synset identifier (previously stored in the DB). In this way all terms belonging to the same synsets (real synonyms or simple term variants) will be considered equivalent in further processing. This approach leads to a degree of normalization for what concerns terminology representation. If a query contains a term, the answers retrieved might contain any of the variants of that term, which are considered equivalent in the semantic representation because they belong to the same synset.

By way of example, a simple sentence such as *the toggle switch controls the cargo compartment doors* produces the MLF:

(9) **holds**(o1),
    object(toggle_switch, o2, [x1]),
    object(cargo_compartment_door,o3,[x2]),
    event(control, o1, [x1,x2]).

Parsing the multi-word terms as single tokens identifies two objects x1 the toggle_switch and x2 the cargo_compartment_door. The final line asserts that there is an event o1 in which x1 controls x2. A characteristic feature of all MLFs is exemplified in (9) by the additional arguments o1, o2 and o3. These are the result of reification and allow further modification of any of the existing predicates without the need to embed arguments, maintaining a functional flat structure (Mollá et al., 2000).

Semantically, considering a term as a synset will replace the actual tokens in (9) with their synset identifier:

(10) **object**(synset_1, o2, [x1]),
     object(synset_2, o3, [x2]),

Where synset_1 contains *toggle switch, toggle switches* and synset_2 contains the terms in (1) and (2). Now any variation in terminology will not prohibit the logical proof criteria for answer extraction. A query looking
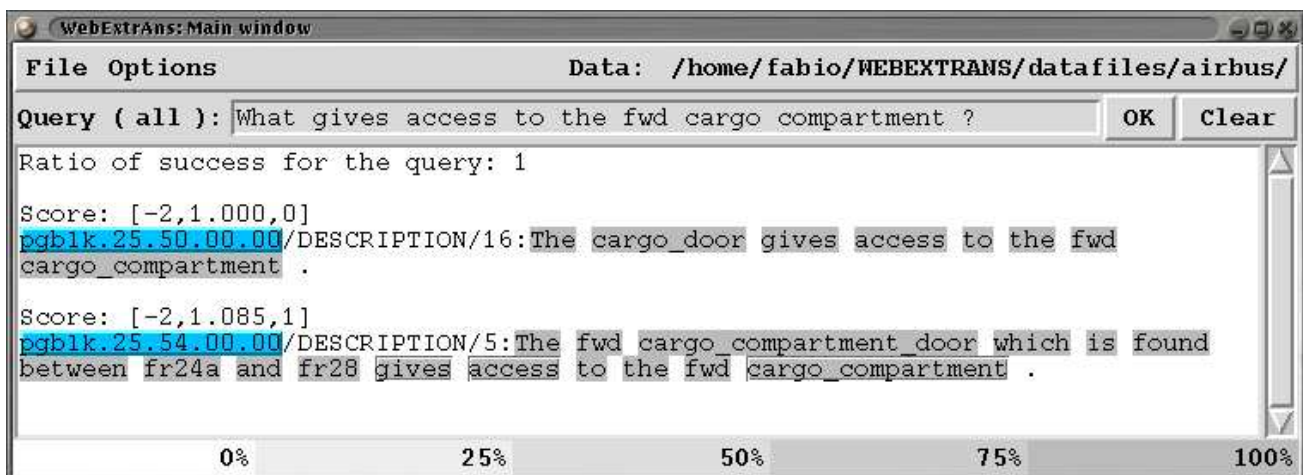
WebExtrAns: Main window

File Options   Data: /home/fabio/WEBEXTRANS/datafiles/airbus/

Query (all): What gives access to the fwd cargo compartment ?   OK   Clear

```
Ratio of success for the query: 1

Score: [-2,1.000,0]
pgb1k.25.50.00.00/DESCRIPTION/16:The cargo_door gives access to the fwd
cargo_compartment .

Score: [-2,1.085,1]
pgb1k.25.54.00.00/DESCRIPTION/5:The fwd cargo_compartment_door which is found
between fr24a and fr28 gives access to the fwd cargo_compartment .
```

0%   25%   50%   75%   100%

Figure 5: Example of interaction with ExtrAns

for (9) can use any of the methods of refering to the *cargo compartment door* that are used in the manual.

6. Conclusion

In this paper we have described the peculiarities of the terminology in a very specific type of document. We have proposed and evaluated combined techniques for Terminology extraction, that take advantage of the specificity of the domain. We have described how the extracted terminology has been exploited in the context of an Answer Extraction system, in particular in form of a domain-specific ontology.

Our goal is to reach a stage where terms will cease to be a computational burden and become the key to unlock meaningful answers in the AE task.

7. References

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

Eric Brill and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proc. COLING '94*, volume 2, pages 998–1004, Kyoto, Japan.

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.

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.

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

John S. Justeson and Slava M. Katz. 1995. Technical terminology: some linguistics properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Michael Louw. 1998. Polaris User's Guide. The EuroWordNet Database Editor. EuroWordNet (LE-4003), Deliverable D023D024. Technical report, Lernout & Hauspie - Antwerp, Belgium.

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

Fabio Rinaldi, Michael Hess, Diego Mollá, Rolf Schwitter, James Dowdall, Gerold Schneider, and Rachel Fournier. 2002. 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.

Piek Vossen. 1997. EuroWordNet: a multilingual database for information retrieval. In *Proceedings of the DELOS workshop on Cross-language Information Retrieval*, Zurich.

Piek Vossen, editor. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, Dordrecht.