RC22615 (W0210-206) October 31, 2002
Computer Science

IBM Research Report

Terminology Extraction for Global Content Management

Arendse Bernth, Michael C. McCord
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Kara Warburton
IBM Software Development Laboratory
8200 Warden Avenue
Markham, ON L6G 1C7
Canada

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Terminology Extraction for Global Content Management

Arendse Bernth, Michael McCord and Kara Warburton

Authors' addresses:

Arendse Bernth and Michael McCord
IBM T. J. Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598
U.S.A.

Kara Warburton
IBM Software Development Laboratory
8200 Warden Avenue
Markham, ON L6G 1C7
Canada

The role of terminology in content management has often been underrated. However, supplying key product terms to translation services several weeks before a translation shipment arrives not only reduces translation time and improves translation quality, but also saves effort (and thus money) by reducing duplication of work by the translation services. Getting the key terms ready in a timely manner can be difficult without some automation. This paper describes the process of proposing, designing, developing, and deploying a terminology extraction tool. The tool extracts nouns and noun combinations, excludes non-translatable terms and known product terms, and displays a context for each extracted item. This is done based on full parsing of the text with a broad-coverage parser. The tool is made available to users on a Web server.

Keywords: Term recognition, multiword terms, terminology extraction, computational-linguistic tools, terminology management.

1. Introduction

Managing information, or "content," to improve product quality and employee productivity has gained focus as companies look for new ways to increase their competitive edge in global markets. Product managers who used to send their books or brochures to their local vendor for publishing or translation are now looking for high-tech solutions to streamline content creation and localization in-house. Content reuse or recycling, text mining, controlled authoring, information retrieval, localization, translation memories, and machine translation — often collectively referred to as *content management* — these are the buzz words of today's information technology professionals.

While the aforementioned strategies gain attention, the role of terminology in content management has often been misunderstood or underrated. Most controlled authoring and translation memory systems, for example, fail to focus on the term as an information unit. While today's authoring and translation tools have evolved to include sophisticated style and spell checkers, translation memories, format converters, and back-end databases for recycling, many continue to offer only basic terminology functions that differ little from

traditional desktop dictionaries. Today, the potential benefits of terminology management are seldom fully realized by enterprise content management systems.

To promote terminology as a key aspect of content management, the Localization Industry Standards Association (LISA), a consortium of globalization and localization industries, recently conducted a survey on terminology management. The survey examined how companies currently manage the use and translation of terms through the production cycle. The survey revealed that terminology extraction is a task that could realize significant benefits; yet few technical applications are available to perform it.

As a follow-up to the LISA survey, IBM conducted a survey among staff involved in the end-to-end production cycle (product developers, writers, translators, user-centered designers, and marketing representatives) to identify ways to improve terminology management in support of global content management objectives. The most pressing need expressed by the translation staff was aligned with the findings of the LISA survey — to provide a tool that can automatically produce a list of key terms in a product that is about to be translated. IBM subsequently embarked on a project to develop such a tool.

This paper describes the process of proposing, designing, developing, and deploying the IBM terminology extraction tool, called TermExt.

2. Concept

After the IBM translation services requested the terminology extraction tool, a workgroup of user representatives was formed to build a business case, define the base requirements and describe how the tool would be used.

Due to global market pressures, the difference between the availability date of the English version of a product and that of its translated versions has a direct impact on sales; the longer this gap, the less revenue will be realized for the translated product. The ultimate goal is simultaneous release, referred to in the industry as “world wide simultaneous general availability.”

Reducing translation time is therefore essential. The thousands of files that need to be translated for a given product are thus usually divided into smaller sets and sent to different translators who work simultaneously on their portion of the files.

Translators use the IBM TranslationManager (TM) authoring software. This software includes translation memory capabilities, whereby if the English sentence has been previously translated, the existing translation can be reused. It also contains a dictionary that stores English and translated terms and other metadata such as definitions. The dictionary automatically displays the recorded translation of an English term when it is encountered in the active translation segment. Often called the “push” approach, this “forces” the selection of consistent translations by different translators who, because they often work in isolation of each other, might otherwise use different terms.

The creation and use of bilingual TranslationManager dictionaries containing the key product terms is therefore essential to achieving consistent, high-quality translations and increasing translator productivity. But within the short schedules, the translation services frequently lack the time to identify the key product terms, record them in the dictionary, and research and record the appropriate translations. Consequently, some translators receive the files without any accompanying dictionary, or the dictionary may be incomplete. Translation time can be extended as translators spend time researching terminology, sometimes even the same terms. Furthermore, without the dictionaries available to suggest consistent terms, inconsistent and incorrect terminology occur in the translated product. These problems are more costly to correct later during Translation Verification Testing (TVT), if they are detected at all.

Even if the translation service does find the time to manually identify the key product terms and add them to the TM dictionary, doing so represents a significant duplication of work since the translation services in other countries are having to do the same task. For a given product, as many as 30 different translation services may be performing manual term extraction from the same set of English files. Some translation services have fewer resources than others to do this task. The incomplete dictionaries in these languages can result in inferior quality translations in those countries.

Generating the terminology list upstream, at the content authoring phase, allows this duplicate work to be avoided. Furthermore, content authors, with their native command of the source language and their extensive product knowledge, know more about what constitutes a key product term than the translator would.

Thus the translation services requested a tool that would supply them with key product terms. If they received the list of terms generated by this tool several weeks *before* the translation shipment arrived, they would have enough time to research and record the translations. When the product arrived for translation, the bilingual dictionary would be ready and could be supplied to all translators. The key benefits: reduced translation time and improved translation quality.

3. Proposal

The IBM translation services were then approached, as part of the process of building a business case, to show how the terminology extraction tool could realize a return on investment.

As is the standard in the translation industry, translations are charged by the word. A key objective is always to reduce costs. But the benefits of the terminology extraction tool are reduced translation time and improved translation quality. Reduced translation costs would not be realized because the same number of words would be translated. Thus a formidable challenge in the business case was to prove the tangible benefits of intangibles such as quality and time. In the competitive race for investment dollars, the terminology extraction tool proposal was heavily scrutinized against other proposals whose business cases showed measurable profits or cost savings. This experience

confirmed yet another finding of the LISA survey: building a business case for terminology management can be quite difficult.

The business case contained an estimate of the time savings that would be realized, and a description of the quality benefits. For continued support of the project, it must be shown that benefits are actually realized, such as through customer satisfaction scores, decreases in TVT error reports relating to terminology, and reduced time-to-market for translated products. Fortunately, IBM's globalization executives realize that product quality and time-to-market are competitive advantages and so they approved the project.

4. Requirements specification

The next decision was to "build, buy, or modify." Should IBM develop a new term extraction tool, purchase one, or modify an existing one to meet the requirements? The Workgroup defined the following base requirements and then conducted a competitive evaluation.

4.1 Extract nouns

An evaluation of existing terminology resources in IBM showed that ninety percent of the terms they contain are nouns. Nouns are most often used to denote key concepts such as product components, features, user interface objects, and so forth. Nouns most frequently require effort to research an appropriate translation and ensure that it is used consistently. Extracting only nouns was therefore accepted as a base requirement. This includes the following types of nouns and noun combinations:

- noun, e.g. *servlet*
- noun + noun^{+}, e.g. *data definition language*
- adjective + noun^{+}, e.g. *flat file*
- past participle + noun^{+}, e.g. *forked children*
- present participle + noun^{+}, e.g. *calling sequence*

4.2 Exclude non-translatable terms

Some text in products must not be translated, such as, in the case of software, names of commands, parameters, variables, entities, and so forth. These types of text are called machine readable information (MRI) because they are intended to be read by software programs as part of the product functions. For example, a command could be integrated into a URL of a web-based product in order to retrieve data from a database and display it on a Web page. These strings often resemble English words, to make them meaningful to and easily remembered by both the product developer and the source language user — for example, the command *SHIPTO* in an online store. If these strings are translated by mistake, the product will fail.

To help prevent them from being translated by mistake, MRI strings should be excluded from the list of terms sent to translators. The workgroup suggested the following approach to achieve this:

1. Since MRI strings are typically non-compound (strings that contain no space), any string of letters without a space that cannot be found in a general purpose dictionary may possibly be a non-translatable string, for example, *parmlib*. However, some such strings could be valid new terms, such as *servlet*. Therefore, rather than eliminating them, they should be extracted into a separate "unknown words" section of the list.
2. Frequently, the product team has a list of MRI strings. They are often, for example, documented in a reference manual. If such a list is available, it can be used as an exclusion list to reduce the size of the list of unknown words produced in step 1.
3. Most document and help files in IBM software products are authored in a standard SGML or HTML format. Usually, MRI strings are preceded and followed by one of a limited number of markup tags, such as *<ph class="IBMcommand">…</ph>*. The tool could be programmed to ignore all text found between any of these tags.
4. On some server platforms, MRI strings are written in uppercase, such as *ALTUSER*. Eliminating uppercase strings would, however, also eliminate acronyms. Since IBM has a large database of known acronyms, it was decided to ignore all uppercase terms found in the product files and separately provide translators with the acronyms from that database.

4.3 Exclude known product terms

Unless a product is in its first release and uses brand-new, cutting-edge technologies that require a lot of new terms, a high proportion of the terms in the product materials are likely to be already available with predetermined translations in the translation services bilingual dictionaries. DB2 UDB, for example, has had seven releases and has already been translated into many languages. This is what is called a "mature," "globalized" product. Likewise, many IBM software products that run on the Windows™ operating system use standard terms which have pre-determined translations in the dictionaries. It would be useful to the translators if the list of terms for a product requiring translation did not contain such words.

L'Homme et al (1999) describe the use of a term extractor that did not automatically eliminate known terms. Ninety percent of the output constituted known terms that had to be identified and eliminated manually, a time-consuming and tedious task.

IBM has a terminology database, maintained in Toronto, Canada, that contains approximately 20,000 English terms with other metadata such as definitions, part-of-speech values, contexts, usage notes, and status indicators (preferred, deprecated, and so forth). The terms in this database originated from glossaries from mature products, such as the *DB2 Glossary*. Therefore, most of these terms in this database are precisely the ones that should already have translations in the translation dictionaries.

Consequently, eliminating known product terms from the terminology list can be achieved by using the terms from the English terminology database as an exclusion list. Since the English terminology database is stored in a highly-structured DB2 system, it is easy to export the desired terms (only nouns with "preferred" status from the so-called "mature" products) in a format that is usable by the terminology extraction tool. This is an instance where having terminology in a well-structured database really pays dividends because the content can be used by machines as well as by humans.

4.4 Display contexts

The term alone is not enough information to determine an appropriate translation, since of course terms can be polysemous. Additional conceptual descriptions are necessary to provide clues as to the meaning of the term. Preparing a definition for each and every term in the list is not feasible due to the number of terms, and for many terms whose meaning is self-explanatory this exercise is unjustified. However, a context, or example sentence, for the term could be extracted automatically. The name of the source file where the context appears in should also be indicated so that the translator can read the macro-context if needed for further clarification.

Contexts have long been used in terminology management to demonstrate meaning, collocations, and usage information. Because of their automated retrievability, contexts are often integrated into terminology management tools. Dubuc (1992) describes three types of contexts: defining contexts, explanatory contexts, and associative contexts, the first being closest to a definition. While it would be best to extract defining contexts, the automatic analysis of contexts was outside of the scope of the project's first release, and so we limited our requirement to full-sentence contexts. This would at least eliminate the extraction of extremely limited contexts such as those found on user interface controls.

5. Competitive evaluation

Six commercially-available terminology extraction products were then evaluated to see if they could meet the following base requirements:

- Extract only nouns and noun combinations
- Eliminate non-translatable (MRI) terms
- Exclude known product terms
- Display contexts

The products were used to extract terms from the same source files under the same extraction conditions, and then the results were compared. The products displayed a range of problems or limitations, summarized below.

- Either only simple terms, or only compounds, are extracted, but not both.
- Translatable terms are indistinguishable from non-translatable strings.
- Uppercase terms are converted to lowercase.
- All parts of speech are exported, instead of just nouns and noun phrases.
- Both the singular and the plural forms of a term are exported.

- Both the noun and the article plus the noun are exported.
- The complete range of additional information is not provided (frequency, context, and source)
- Non-alphabetic characters are not filtered out, such as numbers and symbols.

A major problem was that none of the tools could parse the proprietary SGML and XML file formats that are used to author most of IBM content. Some of the commercially available tools supported only plain text files without markup, and several also supported a standard word processing format such as MS Word. None were sophisticated enough to handle SGML- or XML-compliant proprietary formats. The use of proprietary file formats provides strategic benefits for large companies, such as facilitating information reuse, supporting the development of in-house tools, minimizing the dependency on vendor products, and fostering innovation. However, off-the-shelf tools normally require retrofitting to support proprietary file formats. The cost a vendor would charge to add support of internal file formats and the additional specific functions needed would likely exceed the cost of developing a new tool in-house. Furthermore, IBM has already developed expertise and technologies to process internal file formats that would take more time, and cost, for an external company to master.

IBM had previously developed a terminology extraction tool (Bernth 1997) as part of its document authoring tool, the Information Development WorkBench. This tool already supported IBM file formats and one of the core requirements: extract only nouns and noun combinations. This tool was therefore selected for further development to meet the additional requirements. The tool's developers at the IBM Watson Research Center (the first two authors of this paper) were engaged to complete the work.

6. Technical Implementation

In contrast to e.g. Damerau's (1990) and Church and Hanks' (1989) statistical approaches to term extraction, a central element of TermExt is the *English Slot Grammar* (or *ESG*), a full-fledged English parsing environment (McCord 1980, McCord 1990). ESG handles various text formats, such as HTML, SGML, and plain text. The system segments and tokenizes the input text, performs morphological analysis (including derivational as well as inflectional morphology), and finally assigns syntactic structures to the sentences. The syntactic structures show not only surface relations, but also deeper relations, as exemplified by its treatment of passive constructions and remote relations. A diagram giving an overview of the system architecture of TermExt (with ESG) is given in Fig. 1.

Armed with this rich environment, the developers proceeded to implement the agreed-on design, and in the process contributed to refinements. The basic requirement was to extract various types of noun phrases. ESG has a dictionary of about 85,000 lemmas of general English vocabulary. The number of recognized words is much higher, considering the morphological analysis. This lexical analysis and the deep parse structure form the basis of the term extraction. TermExt goes through the list of nodes in the parse tree, looking for head nouns.

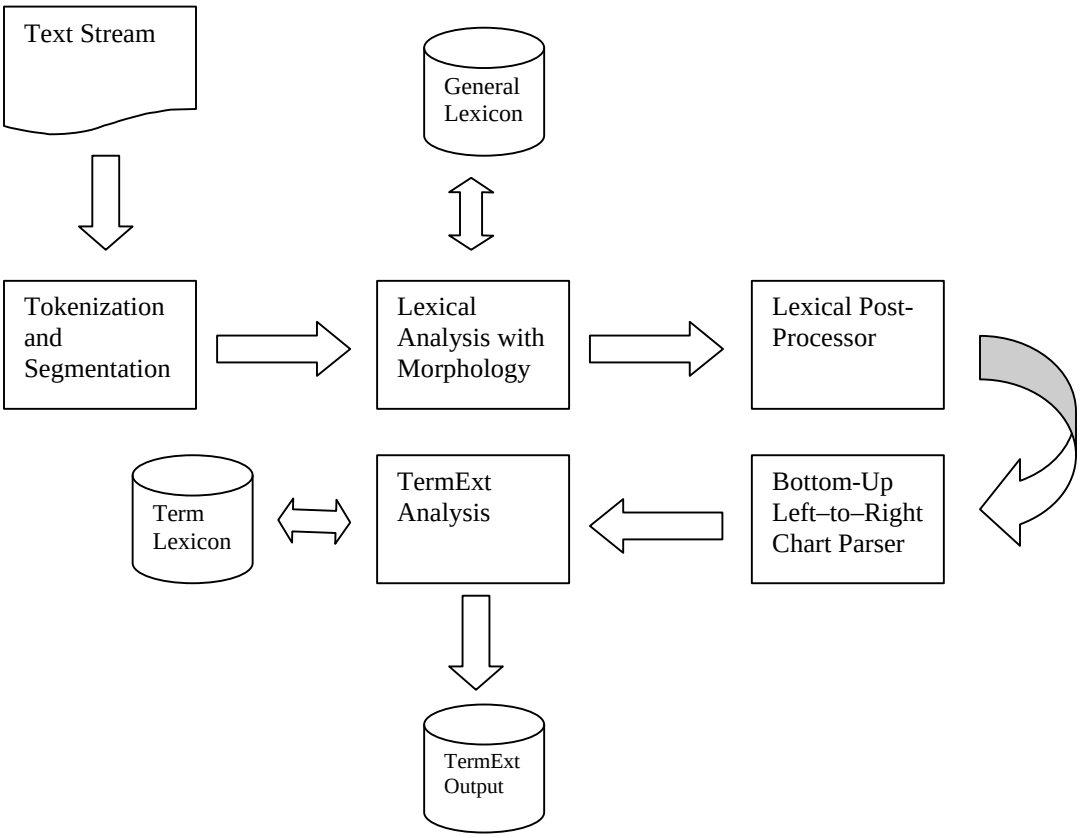

Fig. 1. *TermExt Architecture.*

Once a head noun has been identified, the system looks for the left boundary of the noun phrase. This sets a limit for the leftmost token of the term candidate. Next, the system proceeds from right to left, starting with the head word, including noun pre-modifiers of the required kinds, until either the left limit has been reached or an undesired noun pre-modifier, such as a determiner, has been reached.

The full parses provided by ESG can make term identification more accurate. This is illustrated with the sentence in (1).

(1) At the destination point files are stored.

A simple POS tagger will not identify the scope of the prepositional phrase, "At the destination point", with the result that the extracted term could wrongly include *files*, whereas the ESG parse, shown in Fig. 2, identifies *point* as the head of the prepositional phrase, and *files* as the head of a separate noun phrase.

```
------------------------------------------------------------------------
.----- vprep      at1(1,4)          prep pprefv staticp
| | .- ndet       the1(2)           det sg def the ingdet
| | .- nnoun      destination1(3,u) noun cn sg
| `--- objprep(n) point3(4,u)       noun cn sg locnoun meas lmeas
.----- subj(n)    file1(5,u)        noun cn pl
o----- top        be(6,5,7)         verb vfin vpres pl vsubj
`----- pred(en)   store1(7,u,5,u,u) verb ven vpass sta
------------------------------------------------------------------------
```

Fig. 2. *ESG Parse of* "At the destination point files are stored."

Another example of how full parsing helps identify the correct scope of the noun phrase is shown in (2).

(2) The help users need is provided in the manual.

Here a simple POS tagger will not identify the abbreviated relative clause, "(that) users need", with the result that the extracted term might wrongly include *users,* whereas a full parse identifies *help* as the head of one noun phrase, and *users* as the head of a different noun phrase.

Certain noun pre-modifiers, especially adjectives, pose particular challenges. Some adjectives express properties whose applicability is subject to change, such as *new*, *bad*, and *current*, and their antonyms. These adjectives are not normally part of terms (even though they may be, as exemplified by *new structure*, which is a synonym in a certain domain for *primary group buffer pool*) and hence should not be included in the noun phrases that are extracted as term candidates. We have opted for the simple solution of keeping a hand-coded list of such adjectives to be used as a filter; this is in line with the approach of Heid (1998/1999), who also outlines a method of using an automatically derived collection of general language adjective-noun collocations as a filter. Park *et al.* (2002) take the latter approach of Heid a step further and describe a more complex method for automatically deciding the applicability of the adjective for a given domain. Their approach depends on two measures associated with an adjective. One is the domain-specificity of the adjective; the other is the *association* of the adjective with the noun following it, which is the relative probability of occurrence of that noun, given the adjective. Basically, the adjective is included in the term if these two measures are high enough. A possible drawback of such a method is that computing these measures requires training on the various domains of interest.

By default, ESG treats extralexical tokens (words not found in the lexicon after morphology has applied) as proper nouns; this often allows the segment to parse. These nouns are also put in a list of unrecognized words and output separately for the terminologist to look over, thus fulfilling part of the requirement to exclude potentially non-translatable terms. Further, ESG's ability to handle various markup languages made it possible to look for specific tags that indicate MRI environments, thus fulfilling another part of the requirements. Finally, some filtering of term candidates is achieved by look-up in special lexicons of MRI terms and known product terms.

As Justeson and Katz (1995) rightly point out, conjunctions are very unlikely to occur as part of terms. For this reason, TermExt is called recursively on the left and right conjuncts, whenever a noun phrase whose head is a conjunction is encountered.

For each term candidate, the number of occurrences and the first full-sentence context that it occurred in are output. (If the term occurs in no full-sentence context, then the chosen context is simply the first segment it occurs in.) The context display also shows the name of the file from which the example segment is taken.

TermExt is mainly rule-based, but can be viewed as a hybrid system due to the statistical aspect of producing and using frequency counts for terms; these frequencies can feed into the parsing, which employs numerical scoring. Systems like the ones described in Heid (1998/1999) and Park *et al.* (2002) use more statistical and less linguistic methods. Heid (1998/1999) uses a part-of-speech tagger, and Park *et al.* (2002) use shallow parsing. Both systems use the well-known statistical technique of relative frequency comparison (Edmundson and Wyllys 1961). The central idea of this technique, when applied to term extraction, is to compare the (relative) frequency of a term candidate in a general domain to its frequency in the specific domain that the term extraction applies to. If the frequency in the specific domain is substantially higher than in the general domain, then the hypothesis is that the candidate is more likely a real term. As (Heid 1998/1999) points out, one difficulty of this technique is to obtain a sufficiently large and sufficiently general corpus to train on for the base comparison. His system uses journalistic corpora and so may be biased for terms in the fields of e.g. economy and sports.

Heid (1998/1999) and Pazienza (1998/1999) describe systems that use a bootstrapping process, in which singleton nouns whose distributional properties designate them as potential term candidates are identified in a prior pass. These nouns are then fed into a subsequent pass for identifying multiword terms. Whereas TermExt operates in one pass, it would certainly be possible to use the general addendum lexicon facility to include singleton nouns collected in a prior pass. As it is, TermExt uses the results from prior runs in a more indirect manner, in that terms extracted earlier eventually become part of the terminology dictionary that is used as an exclusion list for known terminology.

As mentioned above, documents are often segmented into many smaller files. Since frequency measures are an important aspect of term collection, it is necessary to be able to run a term extractor on several files and allow it to gather terms and their overall frequencies across these files. TermExt is able to handle vast amounts of files; it proved its robustness when run successfully on 87,000 files in one session.

Fig. 3 shows the TermExt program output. Meta-information is output at the top of the file. That includes the input file name, which could be a file containing a list of files to process, and explanation of the output format. The output is divided up into three sections: Multiword terms, single known words, and unknown words. Within these sections, each entry has a similar structure, *viz.* term, part of speech, frequency, example segment, and file name where the example segment originates from. The "<" character

separates the fields. Each entry is in reality on a separate line; in the figure, we show the lines wrapped around.

```
<<< TermExt Analysis of File Voicec6\*.ide >>>
    Each entry of form:
    Term < Part of Speech < Frequency < Example Segment << File Name

<<< Multiword Terms >>>
phone number < n < 27 < The &VEI; for &dtm; is set up on its own phone
number,so that subscribers can choose between the &VEI; and the telephone
keypad interface. << Ch1.ide
WebSphere Voice Server server component < n < 20 < To achieve good performance,
install the WebSphere Voice Server server components on their own server. <<
Ch1.ide
Main Menu < n < 18 < You have now reached the Main Menu. << Ch4.ide
application server < n < 16 < Create an application server for the &VEI;. <<
Ch2.ide
voice mailbox < n < 14 < To use &dtm;, you need a &dtm; voice mailbox and an
internet e-mail account (residing on an IMAP4-compliant or POP3-compliant e-
mail server). << Ch4.ide

<<< Single Known Words >>>
voice < n < 183 < &dtm; is a unified messaging system that uses the voice
processing capabilities of &prodt; to let you access, process, and send voice
mail, fax, and e-mail messages. << Ch1.ide
system < n < 132 < &dtm; is a unified messaging system that uses the voice
processing capabilities of &prodt; to let you access, process, and send voice
mail, fax, and e-mail messages. << Ch1.ide
file < n < 84 < The &VEI; for &dtm; is supplied with pre-recorded prompts in a
WAV file format. << Ch1.ide

<<< Unknown Words >>>
keypad < unk < 24 < <ph style="bold">Telephone keypad interface:</ph> <<
Ch1.ide
dtuser < unk < 6 < To find out whether you have installed this component on
your &library; system, log on to AIX using the &osq;dtuser&csq; ID. << Ch2.ide
erlang < unk < 4 < The erlang has a very concise meaning for mathematicians and
queuing theory experts. << glossary.ide
```

Fig. 3. *TermExt Output.*

7. Deployment

Ease of installation and use are of utmost importance in order for a new tool and process to be widely accepted. Most employees do not want to install and maintain yet another software program. The tool was therefore set up on a Web server so that it could be launched from a Web site. The user simply goes to the Web site, clicks a button to upload files for analysis, and a few moments later the results appear. The Web interface was designed with input from User Centered Design (UCD) professionals who focus on simplicity and ease-of-use. Consequently, employees can usually run the tool quickly and easily without needing to learn new programs or read user manuals.

The results are delivered in two formats. The first is a plain ASCII list shown in Figure 3, which is provided for convenient browsing of the output. The second is a

TranslationManager dictionary which contains the same information in a proprietary SGML format.

The TranslationManager dictionary is sent to the IBM translation services where it is imported into the TranslationManager. A translation service employee simply adds the appropriate target language translations, using a graphical interface. When the product to be translated arrives, the bilingual dictionary is ready to be provided to the various translators who are involved.

8. Applications and Benefits

Tests on actual source files confirmed our expectation that TermExt would have additional uses and benefits beyond those of translators.

Because TermExt uses comparisons with a general purpose dictionary and the IBM Terminology Database, the unknown words section reveals problems that should be corrected in the source files, such as errors in spelling, capitalization (*boolean*, instead of *Boolean*) and word formation (*data base*, instead of *database*). This section also contains any new words (neologisms). These are evaluated by UCD professionals to ensure that they are appropriate and easily localized, and then the terminologist prepares a definition and records the information in the terminology database. This helps to improve consistency of terminology use across the company.

Running TermExt on the files of different writers at the same time also allows editors to detect inconsistencies between writers. They can re-sort the output alphabetically to align and compare like terms. Excessively long strings of nouns and qualifiers without prepositions (such as *partitioned data set member name*), which are difficult to translate, can be easily identified and reformulated to a more localization-friendly structure.

Thus, the tool is effective for a range of content management tasks: provide key terms to translators for determining target language equivalents, detect spelling and typographical errors, increase standardization of word formation, enhance the localizability of terms, improve style, manage the creation of new terms, and supply a source of new terms for the English terminology database.

9. Future possibilities

As we gain experience using TermExt, we will look for ways to improve it. For example, rather than assuming that the terms in the English terminology database already have translations in the bilingual dictionaries, we can develop routines to directly compare the output with the bilingual dictionaries and remove previously-translated terms. By integrating TermExt directly into the Information Development WorkBench and other development tools, we could automate the process of running TermExt and sending the output to the translation service, thereby making it virtually transparent to the user.

Running TermExt at the earliest design stage of a product's development allows us to initiate a terminology standardization process that helps prevent errors before they occur.

In its current state, TermExt outputs *all* multiwords, known nouns, and unknown words indiscriminately. An obvious improvement would be to apply a statistical filter. Damerau (1993) compares different approaches and concludes that a simple relative frequency comparison is at least as good as more elaborate calculations, so this seems like a good next step in improvements. Another improvement would be to improve the quality and usefulness of the context sentence by looking for defining contexts rather than merely giving the first context found.

10. Conclusion

While we initially embarked on terminology extraction as a way to solve a translation problem, we are realizing that it has applications and benefits in source language content management as well. The next challenge is to have it recognized and fully integrated as an essential component of global content management strategies.

References

Bernth, A. 1997. "EasyEnglish: A Tool for Improving Document Quality". In *Fifth Conference on Applied Natural Language Processing*, 159-165, Washington, DC, USA. Association for Computational Linguistics.

Church, K. W. and P. Hanks. 1989. "Word Association Norms, Mutual Information and Lexicography." *Proceedings of the 27th Annual meeting of the Association of Computational Linguistics*, 76-83, University of British Columbia, Vancouver, Canada, 26-29 June.

Damerau, F. J. 1990. "Evaluating Computer-Generated Domain-Oriented Vocabularies." *Information Processing & Management* 26(6): 791-801.

Damerau, F. J. 1993. "Generating and Evaluating Domain-Oriented Multi-Word Terms from Texts." S *Information Processing & Management* 29(4): 433-447.

Dubuc, R. 1992. *Manuel pratique de terminologie*. Linguatech.

Edmundson, H. P. and W. Wyllys. 1961. "Automated Abstracting and Indexing – Survey and Recommendations." *Communications of the ACM* 4: 226-234.

Heid, U. 1998/1999. "A Linguistic Bootstrapping Approach to the Extraction of Term Candidates from German Text". *Terminology* 5(2): 161-181.

Justeson, J. S. and S. M. Katz. 1995. “Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text.” *Natural Language Engineering* 1(1): 9-27.

L’Homme, Marie-Claude. 1999. “Recherche terminographique semi-automatisée en veille terminologique: expérimentation dans le domaine médical.” *Réseau international de néologie et de terminologie* 20 : 25-35.

McCord, M. C. 1980. “Slot Grammars”. *Computational Linguistics* 6: 31—43.

McCord, M. C. 1990. “Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars”. In R. Studer (ed.), *Natural Language and Logic: International Scientific Symposium*, Lecture Notes in Computer Science, 118-145. Springer Verlag, Berlin.

Park, Y., R. J. Byrd and B. K. Boguraev. 2002. “Automatic Glossary Extraction: Beyond Terminology Identification". In *Proceedings of the 19th International Conference on Computational Linguistics (COLING2002)*, 772-778, Taipei, Taiwan, August 24 - September 1.

Pazienza, M. T. 1998/1999. “A Domain Specific Terminology-Extraction System”. *Terminology* 5(2): 183-201.