Selecting an Appropriate Language Base for Automated Requirements Analysis

In companies that constantly develop new software releases for large markets, new requirements written in natural language continuously arrive that may affect the current development situation. Before making any decision about the requirements, they must be somewhat analysed and understood and related to the current set of implemented and waiting requirements. The task is time-consuming due to the high inflow of requirements and any support that may help the requirements analysts and reduce labour would provide faster and improved decision-support. This paper investigates the terms used in software requirements written in natural language. The investigated requirements come from a real industrial setting and have been used for 5 years when developing subsequent releases of a software application. Through term extraction, sets of requirements from different years, are compared to each other, to the BNC Sampler, and to the documentation of the application. The results show that (1) the language used in requirements may be considered to be unique compared to general spoken and written language, (2) the requirements and the documentation share basically the same domain-specific words, whereas the most used phrases differ, and (3) the language in the requirements is fairly consistent over the years. The results indicate that previous requirements is the better source, compared both to general language and documentation of the software, when building tools based on natural language processing techniques to analyse requirements.

1. Introduction

In software development, requirements are an important means for meeting the needs of the customer. The requirements may be identified and specified either before the specific software is designed, forming a point of departure for the software engineer, or after the software has been tested or used for some time, thus forming a basis for correcting and improving the software and to meet the needs and complaints of the customer.

In traditional software development, also known as bespoke software development, requirements are usually negotiated and agreed upon with a known end user or customer, before development is pursued or finished (Sawyer, 2000). Thus, the devloping company and the customer together identify and specify the requirements. Further, the customer may provide feedback during development to influence which particular requirements should be implemented.

In market-driven software development, there is very limited negotiation with end users and customers (Potts, 1995; Carlshamre, 2002). Rather, requirements are invented and specified in-house. Thus, the developing company relies on employees providing appropriate requirements. The employees are acting as *stakeholders* for different parts of the organisation. The stakeholders include, for example, marketing department, support, and development, that provide requirements of different kinds, such as bug reports, suggestions for new functionality, suggestions for improving already existing functionality, etc.

In order not to miss any good ideas, anyone within the organisation may submit a requirement and requirements are collected continuously during development. The requirements are most often written in natural language and stored in some kind of repository or database. Reading and classifying, i.e. analysing, the requirements are time-consuming, tiresome tasks and the amount of requirements thus easily becomes overflowing and requirements analysis and manangement may therefore entail bottlenecks in the development process.

Different attempts at performing the requirements analysis more or less automatically have been undertaken (Natt och Dag, Regnell, Carlshamre, Andersson, & Karlsson, 2002, pp. 21–22). These automated techniques could support the requirements analyst by suggesting duplicates and groupings based on the requirements' linguistic content. To perform a deeper linguistic analysis of software requirements, an adequate lexicon is crucial. The lexicon must be domain-specific and also contain everyday vocabulary. Such a lexicons is usually constructed manually since there is, to our knowledge, no other way available today. This makes the techniques rather expensive and less appealing to the software development company. In particular, the market-driven organisation does not interpret requirements the same way as in customer-specific development. In the latter, developers usually have very good domain expertise, whereas the former more often rely on consultants and

Johan Natt och Dag, Dept. of Communication Systems, Lund University, johan.nattochdag@telecom.lth.se
Kerstin Lindmark, Dept. of Linguistics, Lund University, kerstin.lindmark@ling.lu.se

brainstorming sessions (Lubars, Potts, & Richter, 1993). Therefore, it is even more troublesome to manually construct an appropriate domain-specific lexicon.

This paper explores the possibility of constructing a domain-specific lexicon more or less automatically. It is hypothesised that requirements or documentation already available in the organisation may be used as a language base when extracting terms for constructing an appropriate lexicon. The properties of three possible sources for term extraction are investigated. Firstly, the BNC Sampler, which is a two percent sample of the full British National Corpus representing both written and spoken language (one million words per category). Secondly, the documentation of a software application developed by a market-driven software development company. Thirdly, the requirements from the development of several releases of the above-mentioned software application.

This paper is organised as follows: In Section 2, the data sources used in the analysis are described. In Section 0, the techniques used are explained, and how the anlysis was carried out is elaborated in Section 4. In Section 5, the raw results are presented and, finally, in Section 6, the conclusions from the results are presented.

2. Data sources

Three data sources have been analysed and compared, with respect to the vocabulary used:

- *The BNC Sampler*
  The British National Corpus (Burnard, 2000) is a 100 million word corpus of modern British English, both spoken and written. It was created in 1994 by a consortium of leading dictionary publishers and research institutions. From the full BNC, a two percent sample has been created and called the *BNC Sampler*. The BNC Sampler consists of one million words spoken, and one million words written. More information is available at http://info.ox.ac.uk/bnc.

- *Software documentation*
  The documentation (Telelogic Tau 4.2 Documentation, 2001) is written in regular English and should also comprise domain specific words. The documentation used belongs to version 4.2 of the software application and is thus rather mature. In total, it comprises 1,069,478 words (called tokens), of which 14,085 are unique (called types). In Table 1 the number of words of different lengths are shown.

Table 1. Number of words of different lengths in the software documentation.

1-letter words	64,565
2-letter words	171,960
3-letter words	217,433
4-letter words	166,958
5-letter words	94,492
6-letter words	95,828
7-letter words	81,950
8-letter words	56,400
9-letter words	54,228
10-letter words	28,054
11-letter words	18,960
12-letter words	6,935
13-letter words	5,008
14(+)-letter words	2,825
Total	**1,065,596**

The totals differ when counting all words and summing up the counts of different word lengths. The software used (WordSmith) does not explain this, nor has a plausible explanation been found. Fortunately the difference is relatively small (4,118 words, which comprise a 0,36 % difference) compared to the full set.

- *Software Requirements*
  From the software vendor we received a database comprising 1,932 requirements written in natural language. The majority of the requirements are written in English, irrespective of the authors' mother tongue. Thus, the quality varies and in some requirements there are also Swedish phrases. Example requirements, with all the attributes the company uses, can be found in Table 3a and Table 3b.

  Due to the continuous elicitation, the requirements concern different stages of development, such as elicitation, selection, and construction. Thus, the requirements are analysed to various degrees. The requirements have been collected during five years of development, starting in 1996. In Table 2 the number of requirements

Johan Natt och Dag, Dept. of Communication Systems, Lund University, johan.nattochdag@telecom.lth.se
Kerstin Lindmark, Dept. of Linguistics, Lund University, kerstin.lindmark@ling.lu.se

Table 3a. Example requirement submitted 1996.

RqId	RQ96-270
Date	
Summary	Storing multiple diagrams on one file
Why	It must be possible to store many diagrams on one file. SDT forces to have 1 diagram per file. It's like forcing a C programmer to have not more than one function per file... The problem becomes nasty when you work in larger projects, since adding a procedure changes the system file (.sdt) and you end up in a mess having to "Compare systems".
Description	Allow the user to specify if a diagram should be appended to a file, rather than forcing him to store each diagram on a file of its own.
Dependency	4
Effort	4
Comment	This requirement has also been raised within the multiuser prestudy work,but no deeper penetration has been made. To see all implications of it we should have at least a one-day gathering with people from the Organizer, Editor and InfoServer area, maybe ITEX? Här behövs en mindre utredning, en "konferensdag" med förberedelser och uppföljning. Deltagare behövs från editor- och organizergrupperna, backend behövs ej så länge vi har kvar PR-gränssnittet till dessa.
Reference	
Customer	All
Tool	Don't Know
Level	Slogan
Area	Editors
Submitter	x
Priority	3
Keywords	storage, diagrams, files, multi-user
Status	Classified

Table 3b. Example requirement submitted 997. This is actually a duplicate of the requirement in Table 3.

RqId	RQ97-059
Date	Wed Apr 2 11:40:20 1997
Summary	A file should support storing multiple diagrams
Why	ObjectGeode has it. It's a powerful feature. It simplifies the dayly work with SDT. Easier configuration management. Forcing one file for each procedure is silly.
Description	The SDT “Data model" should support storing multiple diagram on one file.
Dependency	4
Effort	1-2
Comment	Prestudy needed
Reference	http://info/develop/anti_og_package.htm
Customer	All
Tool	SDT SDL Editor
Level	Slogan
Area	Ergonomy
Submitter	x
Priority	3: next release (3.3)
Keywords	diagrams files multiple
Status	Classified

from each year is shown together with the number of words they comprise.

Table 4. Number of requirements from the different years and the number of words they comprise.

Year	Requirements	Words
1996	459	34,588
1997	714	60,944
1998	440	40,099
1999	239	20,029
2000	80	7,911
Total	**1,932**	**163,571**

3. Term extraction

The process of establishing a terminology for a given domain involves not only a meticulous semantic analysis of terms and definition writing, but also extraction of terms from a relevant material (Suonuuti, 2001). Traditionally, this is a time-consuming business, involving a lot of manual work.

Whereas a terminologist normally spends a lot of time reading the material and trying to figure out which words are typical for the domain, we decided to adopt a more mechanical appoach. In this paper, corpus lingustic analysis methods are used to find the words and phrases which could be considered as domain specific terms, and also general language expressions that appear to be used in a specific way, or simply to be over-represented.

Johan Natt och Dag, Dept. of Communication Systems, Lund University, johan.nattochdag@telecom.lth.se
Kerstin Lindmark, Dept. of Linguistics, Lund University, kerstin.lindmark@ling.lu.se

Corpus lingustic analysis methods involve the following:

- The texts to be analysed are collected into one or several corpora.

- Appropriate software tools, such as Corpus Work Bench, QWICK, SARA and WordSmith, are used to produce different statistics of the corpora:
  - *Word frequency lists*
  - *Keywords* (comparing frequency lists)
  - *Concordances* (contexts in which a given word occurs)
  - *n-grams* (strings of *n* words)
  - Lengths and ratios of different units within the corpora (such as sentences, paragraphs, etc.)

In this paper, WordSmith was used to perform all the analyses, as it provides fast and accurate tools for processing large corpora. The statistics presented in this paper include word frequency lists, keywords, and n-grams. Keywords may require some explanation as these have been used differently to measure similarity between corpora. In WordSmith a keyword is defined as follows:

"a) it occurs in the text at least as many time as the user has specified as a *minimum frequency*.

b) its frequency in the text when compared with its frequency in a reference corpus is such that the statistical probability as computed by an appropriate procedure is smaller than or equal to a p value specified by the user."

(WordSmith Tools Help)

The "key-ness" of a word in the corpus is determined by calculating the frequencies and the number of words in both corpora. These are then cross-tabulated and two statistical tests of significance are computed: a χ^2 test (Siegel & Castellan, 1988) and Ted Dunning's Log Likelihood test (Dunning, 1993). The latter gives a better estimate of keyness, especially when comparing long texts or whole genres against reference corpora. Other measures of similarity, using word frequency lists, are discussed by Kilgarriff (1997).

4. Analysis

From the requirements corpus a simple frequency list was first produced. This list was compared with the frequency list from the BNC Sampler corpus, using the Keywords feature in the WordSmith Tools. The resulting list of 500 words, whose frequency in the requirements word list is significantly higher than in the BNC Sampler word list, was then scrutinised manually.

The same procedure was repeated with, in turn, the documentation vs. the BNC Sampler list, the requirements vs. the documentation, subsets of requirements (the requirements from a certain year were extracted as a subset) vs. the whole set of requirements, the subsets vs. each other, each subset vs. a set consisting of the other subsets, and, for each of these comparisons, a comparison in the opposite direction, i.e. the documentation vs. the requirements and so forth.

It is, of course, to be expected that quite a number of the terms used in the requirements are multiwords. To investigate this, bigrams, trigrams, and tetragrams were extracted. This was done using Perl scripts. The amount of multiwords yielded by simply extracting strings of two, three or four words obviously exceeded manageable numbers by far. To clean up, different scripts were constructed to sort out unique occurrences, delete lines starting by numbers, punctuation marks, parentheses and the like. Finally, lines starting with prepositions and conjunctions were deleted.

The number of multiwords retained is still very large, and the next step would be part-of-speech-tagging the corpus so that the set of term candidates can be limited to strings containing at least one noun.

At this stage we decided to try out the *n*-gram finding feature in the WordSmith Tools instead, called "clusters". As a default, this function finds *n*-grams with a frequency of at least two, and since this turned out to give neat, manageable lists of apparently relevant words and phrases, we decided to use these. The lists of bi-, tri- and tetragrams were compared to each other with the keyword feature in a way corresponding to what was done with the single words.

One of the crucial questions for this investigation is whether the terminology of the requirements is consistent throughout the different sets, making it possible to "predict the linguistic content" of the whole set of requirements based on an analysis of a subset, and, even more important, of new requirements based on an analysis of the old ones. To investigate this, the requirement were divded into subsets, according to year of origin. All

Johan Natt och Dag, Dept. of Communication Systems, Lund University, johan.nattochdag@telecom.lth.se
Kerstin Lindmark, Dept. of Linguistics, Lund University, kerstin.lindmark@ling.lu.se

possible different combinations of subsets were analysed. In brief, comparisons of one subset to the rest of the requirements yield a small number of keywords; in the case of bi-, tri-, and tetragrams even none, in quite a few cases. The small number can be even further reduced, if only words that could be term candidates are retained. As for the comparison of one subset with a set consisting of all the other sets, i.e. the set under investigation not included itself, this yields a somewhat larger number of keywords; approximately thirty per set.

A complete overview of all the comparisons that were found relevant for the questions and conclusions in this paper is found in Table 5. The relevance was decided upon afterwards. The other comparisons made were used to obtain an understanding of the nature of the corpora and how they are related to each other. The complete resulting data for the analysis can be found in an Excel workbook at http://www.telecom.lth.se/Personal/johannod/education/cling/AnalysisResults.xls.

Table 5. Overview of comparisons made between the different corpora and subsets of the corpora.

Id	Corpus under investigation	Reference corpus
req→bnc	All requirements	BNC Sampler
req→doc	All requirements	Documentation
doc→bnc	Documentation	BNC Sampler
2req→2bnc	Bigrams All requirements	Bigrams BNC Sampler
4req→4doc	Tetragrams All requirements	Tetragrams Documentation
97→96	Requirements 1997	Requirements 1996
98→96-97	Requirements 1998	Requirements 1996-1997
99→96-98	Requirements 1999	Requirements 1996-1998
00→96-99	Requirements 2000	Requirements 1996-1999
96→Rest	Requirements 1996	All requirements but 1996
97→Rest	Requirements 1997	All requirements but 1997
98→Rest	Requirements 1998	All requirements but 1998
99→Rest	Requirements 1999	All requirements but 1999
00→Rest	Requirements 2000	All requirements but 2000

Complete resulting data can be found at
http://www.telecom.lth.se/Personal/johannod/
education/cling/AnalysisResults.xls

5. Results and anlysis

Already the keywords list resulting from the comparison of the full set of requirements and the BNC Sampler shows some interesting features of the linguistic contents of these texts (Table 6, left column). Among the most significant words are not only domain-specific terms, but overrepresented words with a high frequency in general language, such as *should*, *be,* and *is* (i.e. these are not considered terms at all. The words are boldfaced in the table and those that are adjectives are also italicized). This is not very surprising, considering that the requirements are about features that are not working or features that the requirement stakeholder wishes to have. The comparison between the requirements and the documentation points even clearer in the same direction, where *I, should, would, etc.,* are overrepresented words in the requirements (Table 6, right column).

Table 6. Truncated keywords lists from single word comparison between all the requirements, on the one hand, and the BNC Sampler and the documentation, respectively, on the other hand.

N	req→bnc	req→doc
1	SDL	**I**
2	SDT	**SHOULD**
3	**SHOULD**	**WOULD**
4	SYMBOL	SDT
5	FILE	**IT**
6	MSC	**WE**
7	EDITOR	**TO**
8	ORGANIZER	TODAY
9	USER	**OUR**
10	DIAGRAM	**HAVE**
11	FILES	CUSTOMERS
12	SYMBOLS	ITEX
13	CODE	DOCUMENTATION
14	TEXT	SUPPORT
15	ITEX	**LIKE**
16	**BE**	USER
17	MENU	MINISYSTEM
18	DIAGRAMS	***NICE***
19	SIMULATOR	**THINK**
20	PAGE	**DON'T**
21	DIALOG	**ABLE**
22	TOOL	**MAKE**
23	***POSSIBLE***	**VERY**
24	TAU	***EASIER***
25	**IS**	***BETTER***

The overrepresentaion of expressions common in general language is shown even more clearly in the tetragram comparison (Table 7).

Johan Natt och Dag, Dept. of Communication Systems, Lund University, johan.nattochdag@telecom.lth.se
Kerstin Lindmark, Dept. of Linguistics, Lund University, kerstin.lindmark@ling.lu.se

Table 7. Truncated keywords lists from tetragram comparison between all the requirements and the documentation.

N	4req→4bnc
1	SHOULD BE POSSIBLE TO
2	IT SHOULD BE POSSIBLE
3	TO BE ABLE TO
4	ALLOW THE USER TO
5	IT WOULD BE NICE
6	SHOULD BE ABLE TO
7	IN THE DRAWING AREA
8	THERE SHOULD BE A
9	IS NOT POSSIBLE TO
10	IN THE SDL EDITOR
11	IT IS NOT POSSIBLE
12	MAKE IT EASIER TO
13	WOULD BE NICE TO
14	DEFECT POSTPONED IN #
15	PM DEFECT POSTPONED IN
16	WANT TO BE ABLE
17	IN THE MSC EDITOR
18	THE USER WANTS TO
19	SHOULD BE EASY TO
20	THE USER HAS TO
21	TO MAKE IT EASIER
22	IT IS POSSIBLE TO
23	I WOULD LIKE TO
24	LIKE TO BE ABLE
25	END # #
26	BE POSSIBLE TO USE
27	BE NICE TO HAVE
28	WOULD LIKE TO HAVE
29	THE PROBLEM IS THAT
30	MAKE IT POSSIBLE TO

By comparing the tetragrams from the requirements to to those in the documentation, it can be further established that the differences are not due to the particular domain, but rather to the type of text. The results from the comparison is shown in Table 8.

Table 8. Truncated keywords lists from tetragram comparison between all the requirements and the documentation.

N	4req→4doc
1	SHOULD BE POSSIBLE TO
2	IT SHOULD BE POSSIBLE
3	TO BE ABLE TO
4	IT WOULD BE NICE
5	I WOULD LIKE TO
6	SHOULD BE ABLE TO
7	ALLOW THE USER TO
8	WOULD BE NICE TO
9	THERE SHOULD BE A
10	DEFECT POSTPONED IN #
11	WOULD LIKE TO HAVE
12	PM DEFECT POSTPONED IN
13	MAKE IT EASIER TO
14	WANT TO BE ABLE
15	TO MAKE IT EASIER
16	LIKE TO BE ABLE
17	THE PROBLEM IS THAT
18	BE NICE TO HAVE
19	END # #
20	IT WOULD BE VERY
21	WOULD LIKE TO BE
22	ID WAS # #
23	PREVIOUS ID WAS #
24	IT SHOULD BE EASY
25	WOULD BE NICE IF
26	I WANT TO BE
27	SHOULD BE EASY TO
28	USER SHOULD BE ABLE
29	THE USER WANTS TO
30	BE POSSIBLE TO USE

As for the actual domain-specific terms, these appear rather from the mono- and bigram lists (and, although to a lesser extent, also from the trigram lists, not shown here). In Table 9, the results from comparing single words lists of the requirements and the documentation, respectively, with the BNC Sampler are shown. As shown in Table 10, by comparing bigrams, domain-specific compound terms may be captured, although there are many non-useful combinations.

Table 9. Truncated keywords lists from single word comparison between all the requirements and the documentation, respectively, on the one hand, and the BNC Sampler, on the other hand.

N	req→bnc	doc→bnc
1	SDL	SDL
2	SDT	TELELOGIC
3	**SHOULD**	TAU
4	SYMBOL	PAGE
5	FILE	TYPE
6	MSC	FILE
7	EDITOR	USER'S
8	ORGANIZER	TTCN
9	USER	C
10	DIAGRAM	MANUAL
11	FILES	**MARCH**
12	SYMBOLS	UM
13	CODE	CHAPTER
14	TEXT	SUITE
15	ITEX	TEST
16	**BE**	SIGNAL
17	MENU	SYMBOL
18	DIAGRAMS	**IS**
19	SIMULATOR	DIAGRAM
20	PAGE	MENU
21	DIALOG	SYSTEM
22	TOOL	PROCESS
23	***POSSIBLE***	CODE
24	TAU	NAME
25	**IS**	COMMAND

Johan Natt och Dag, Dept. of Communication Systems, Lund University, johan.nattochdag@telecom.lth.se
Kerstin Lindmark, Dept. of Linguistics, Lund University, kerstin.lindmark@ling.lu.se

Table 10. Truncated keywords lists from bigram comparison between all the requirements and the BNC Sampler.

N	2req→2bnc
1	**SHOULD BE**
2	**THE USER**
3	**THE ORGANIZER**
4	**POSSIBLE TO**
5	**THE SDL**
6	**THE TEXT**
7	**IT SHOULD**
8	**BE POSSIBLE**
9	**THE MSC**
10	**THE SIMULATOR**
11	**IN THE**
12	**TO USE**
13	**BE ABLE**
14	MACRO MINISYSTEM#
15	**AN SDL**
16	**IS NOT**
17	**THE SYMBOL**
18	SDL SYSTEM
19	**THE EDITOR**
20	**ABLE TO**
21	**THE FILE**
22	**THE SDT**
23	**IN SDT**
24	**IN SDL**
25	**POSSIBILITY TO**
26	**USER TO**
27	SDL EDITOR
28	CODE GENERATOR
29	**THE TOOL**
30	**SDT #**

Several questions were asked before starting out with the analyses, of which two may now be partly answered:

1. *Is there a specific language for requirements?*
   Yes. Firstly, the keywords list compared to the BNC Sampler comprises many domain-specific words (Table 6). Although this is not surprising, the result thus validates the expected. Secondly, the tetragram comparison (between requirements and the BNC Sampler) shows that certain phrases are used more often in requirements than in general language (Table 7). Thirdly, these phrases are not due to the domain, but, rather, to the nature of requirements (Table 8).

2. *Does the language differ or overlap between the documentation and the requirements?*
   Overlap, with respect to domain-specific terms. Differ, with respect to other lingustic features. Comparing both the keywords lists of the requirements and of the documentation to the BNC Sampler shows that the same domain-specific terms occur in both the requirements and the documentation (Table 9). The tetragram comparison between requirements and the documentation has, as already discussed above, shown that the phrases are due to the nature of requirements. Thus, requirements differ in the use of phrases (Table 8).

As for the question of the consistency of requirements throughout the different sets, some relevant results are shown in Table 11 and Table 12. Table 11 list the terms that are significantly more and less common, respectively, in the requirements from 1996 compared to those from 1997. Table 12 shows the corresponding results from comparing the requirements from 1996 to those from all the other years.

It can be seen that some terms are unique for the 1996 set (terms marked with an asterisk). However, most of them may be disregarded as domain-specific terms. It may be concluded from these lists that the words used in requirements are quite consistent over the years.

A very interesting result is that the lists are relatively short, indicating that the differences between the words used in the requirements from 1996 and the words used the subsequent years are limited.

Table 11. Complete list of keywords that are significantly more common in one set or the other.

N	More common 1997 than 1996	More common 1996 than 1997
1	SYMBOL	
2	CMICRO	
3	PAGE	
4	HMSC*	
5	TEXT	
6	SHOULD	
7	FLOW	
8	MSC	
9		ENV
10		ITEX
11		SEND
12		SPEC
13		N

* No occurence 1996

Johan Natt och Dag, Dept. of Communication Systems, Lund University, johan.nattochdag@telecom.lth.se
Kerstin Lindmark, Dept. of Linguistics, Lund University, kerstin.lindmark@ling.lu.se

Table 12. Complete list of keywords that are significantly more common in one set or the others.

N	More common 1996 than other years	More common other years than 1996
1	MINISYSTEM*	
2	MACRO	
3	N	
4	**LNO***	
5	AB	
6	SPEC	
7	CALLER*	
8	SDL	
9	SEND	
10	SHALL	
11	ANSWER	
12	**INMPOVEMENT***	
13	ACTIVEX*	
14	ROOT	
15	SDT	
16	TOOLBARS*	
17	RECEIVE	
18	**AR***	
19	**THANK***	
20	TIMER	
21	AWARE	
22	GATES	
23	COMMAND	
24	DEPENDENCY	
25	OOA	
26	SIMUI	
27		TELELOGIC
28		H
29		FILES
30		TEXT
31		PAGE
32		TAU
33		SYMBOL

* No occurence other years

6. Conclusions

In this paper we have investigated the possibility of constructing a domain-specific lexicon more or less automatically. Based on the analysis in Section 5, it may be concluded that this is possible if using the appropriate language source. Three language sources were compared, software requirements, software documentation, and the BNC Sampler. It was found that:

1. The language used in requirements may be considered to be unique compared to general spoken and written language.
2. The requirements and the documentation share basically the same domain-specific words, whereas the most used phrases differ.
3. The language in the requirements is fairly consistent over the years.

Thus, it may be concluded that:

1. The most appropriate language base for automatically constructing a lexicon would primarily be existing requirements.
2. An initial subset of requirements may be adequate to cover the language used when specifying requirements.

Further investigation on how the lexicon may be constructed is now of great interest. It is suggested that this may be aided by automated taggers, enabling a classification of terms and further automated, yet relevant, term extraction.

7. References

Carlshamre, P. (2002). *A Usability Perspective on Requirements Engineering – From Methodology to Product Development* (Dissertation No. 726). Linköping: Linköping University, Linköping Studies in Science and Technology.

Burnard, L. (Ed.) (2000). *The British National Corpus Users Reference Guide*. Oxford: Oxford University Computing Services, British National Corpus.

Dunning, T. (1993). Accurate Methods for the Satistics of Surprise and Coincidence. *Computational Linguistics, 19*, 61–74.

Kilgarriff, A. & Salkie, R. (1997). Using Word Frequency Lists to Measure Corpus Homogeneity and Similarity between Corpora. In *Proceedings of the Fifth ACL Workshop on Very Large Corpora*. New Brunswick, NJ: Association for Computational Lingustics.

Lubars, M., Potts, C., & Richter, C. (1993). A review of the state of the practice in requirements modelling. In *Proceedings of IEEE International Symposium on Requirements Engineering* (pp. 2–14). Los Alamitos, CA: IEEE Computer Society Press.

Natt och Dag, J., Regnell, B., Carlshamre, P., Andersson, M., & Karlsson, J. (2002). A feasibility study of automated natural

language requirements analysis in market-driven development. *Requirements Engineering, 7*, 20–33 .

Potts, C. (1995). Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software. In *Proceedings of the Second IEEE International Symposium on Requirements Engineering* (pp. 128–130). Los Alamitos, CA: IEEE Computer Society Press.

Sawyer, P. (2000). Packaged Software: Challenges for RE. In *Proceedings of Sixth International Workshop on Requirements Engineering: Foundation for Software Quality* (pp. 137–142). Essen, Germany: Essener Informatik Beiträge.

Siegel, S., & Castellan, N. J (1988). *Nonparametric Statistics for the Behavioral Sciences* (2nd ed.). New York: McGraw-Hill.

Suonuuti, H. (2001). *Guide to Terminology* (2nd ed.). Helsinki, Finland: Tekniikan sanastokeskus.

Telelogic Tau 4.2 Documentation (2001). Malmö, Sweden: Telelogic AB.