Optimal-Constraint Lexicons for Requirements Specifications

Stephen Boyd[1], Didar Zowghi[2], and Vincenzo Gervasi[3]

[1] Softability Pty Ltd & University of Technology Sydney, Australia
[2] University of Technology Sydney, Australia
[3] University of Pisa, Italy
sboyd@softability.com.au
didar@it.uts.edu.au
gervasi@di.unipi.it

Abstract. Constrained Natural Languages (CNLs) are becoming an increasingly popular way of writing technical documents such as requirements specifications. This is because CNLs aim to reduce the ambiguity inherent within natural languages, whilst maintaining their readability and expressiveness.

The design of existing CNLs appears to be unfocused towards achieving specific quality outcomes, in that the majority of lexical selections have been based upon lexicographer preferences rather than an optimum trade-off between quality factors such as ambiguity, readability, expressiveness, and lexical magnitude.

In this paper we introduce the concept of 'replaceability' as a way of identifying the lexical redundancy inherent within a sample of requirements. Our novel and practical approach uses Natural Language Processing (NLP) techniques to enable us to make dynamic trade-offs between quality factors to optimise the resultant CNL. We also challenge the concept of a CNL being a one-dimensional static language, and demonstrate that our optimal-constraint process results in a CNL that can adapt to a changing domain while maintaining its expressiveness.

1 Introduction

Eliminating the ambiguity inherent within a requirement specification is the seemingly unattainable ambition of the systems engineering zealot. This is because ambiguity is characteristic of poor quality requirements, and poor quality requirements are characteristic of challenged projects [1]. It has been suggested that the ambiguity of a requirement can be reduced if the lexicon and/or grammar used to express the requirement is constrained to a subset with stronger properties [2][3]. A Constrained Natural Language (CNL) is a subset of a Natural Language (NL) that has been restricted with respect to its grammar and/or lexicon [3]. By restricting the grammar, complicated sentence structures can be simplified. By restricting the lexicon, unnecessary linguistic variations can be removed, and retained words can be less ambiguously defined.

One of the biggest criticisms of CNLs is that they tend to be unnatural to read and write [4]. Goyvaerts [5] claims that writing requirements in controlled languages is

P. Sawyer, B. Paech, and P. Heymans (Eds.): REFSQ 2007, LNCS 4542, pp. 203–217, 2007.

20% more time consuming that writing requirements in unrestricted NLs. Somers [4] highlights the importance of involving domain authors in all stages of CNL development to ensure the resultant lexicon is natural w.r.t. the domain of interest.

There is a tendency to assume that reduced expressiveness is an unavoidable consequence of constraining a NL. This is because the expressiveness of a language is a measure of the variety of lexical and grammatical constructions it allows [4]. Since a CNL constrains such lexical and grammatical constructions – the subsequent expressiveness of the language is expected to decline. Moreover, existing CNLs are static languages that cannot adapt to express words that have not been designated in advance. CNLs are typically derived from large samples of naturally occurring text in a particular domain [6]. In many cases, a combination of domain experts and automatic parsers are used to extract domain keywords and reoccurring phrases respectively [7]. Fundamentally, this implies that a typical CNL is specific to a particular domain [7][8], and is largely driven by the lexicographers preference. Furthermore, there is a lack of evidence in the literature to confirm whether or not the design of existing CNLs has been rigorously focused upon achieving specific quality outcomes such as unambiguity, readability, and expressiveness.

In this paper we present our fully automatable approach to optimally-constraining the lexicon of a CNL. Our approach aims to exploit important semantic relationships between the words in a requirements sample as a way of logically reducing a NL to achieve a desired level of language quality. We propose a new concept called 'replaceability' which builds upon an existing concept of 'similarity'. We also show how our CNL lexicon remains able to adapt to accommodate new lexical terms that are encountered post its design.

This is significant because existing CNLs tend to be the static result of lexicographer analysis. It is not clear how an existing CNL would be adapted to a new domain – or even how it could be expanded to accommodate a larger sample of text from the same domain. It is of course unlikely that the original lexicographers would always be available to extend their original analysis – and even if they were, it is unlikely that the results would be consistent. On the other hand, we are proposing a new application for existing and well-understood Natural Language Processing (NLP) techniques that practically eliminates the need for a lexicographer in the design of a CNL.

2 Optimal-Constraint Process – Design Goals

There are two fundamentally different constraints underlying any CNL. Firstly there is the constraint on the words that constitute each part of speech (the lexicon), and secondly there is the constraint on the grammatical constructions that will be allowable in the language. The focus of this paper is on optimally-constraining the lexicon. We do not address the issue of constraining the grammar.

Three design goals have been selected to optimally constrain the lexicon – that is to be *readable*, *sufficiently expressive*, and *unambiguous*. Our objective is to achieve the perceived advantage of CNLs (reduced ambiguity), whilst also attempting to overcome the perceived disadvantages (reduced readability and reduced expressiveness).

2.1 Design Goal #1: To Be Readable

A popular criticism of CNLs is that they are unnatural to read and write [4]. Swaffar [9] suggests that what makes text readable is that it “deals with topics of interest or familiar to the intended readers (so that it allows for communication and expressions from within readers' frame of reference).” When a lexicon is constrained, it is unlikely that all words from within the readers' frame of reference will be contained within the constrained lexicon. Consequently, readability is expected to decline.

We believe that when constraining a language there must be cognisance paid to the inclusion of words from the readers' frame of reference. Since we are proposing to derive the CNL from a corpus of existing requirements within the domain of interest, the readers' frame of reference should present itself within the text. For example, by counting the frequency of each disambiguated word within the sample, we gain some insight into the popularity of certain words to express certain meanings. We can then use this insight to help ensure that conventional terms are retained and unconventional terms are replaced within the CNL.

2.2 Design Goal #2: To Be Sufficiently Expressive

It appears that there are two fundamentally different schools of thought on the concept of expressiveness. Gnesi et al [10] and Fabbrini et al [11] imply that expressiveness relates to the ability of a language to convey meaning to a human reader, whereas Nyberg et al [4] believe that expressiveness of a language is some measure of the variety of lexical and grammatical constructions it allows (irrespective of the reader). Here we have two different measures of the size of a language – one relates to the number of semantic meanings that can be generated by a language, whereas the other relates to the number of syntactic expressions that can be generated (which is normally infinite since most useful grammars allow recursion).

Figure 1 shows that a *CNL* consists of a *grammar* and a lexicon of ‘L’ *words*. The *grammar* consists of ‘n’ grammatical *rules* that apply to its eight main parts of speech (*POS*) [13]. Each *POS* consists of ‘W’ *words,* with each *word* having ‘P’ *meanings*. The *CNL* can generate ‘E’ *expressions,* with each *expression* having ‘M’ *meanings* as interpreted by the ‘n’ *stakeholders*. The *domain* of interest is scoped by ‘R’ *requirements.* Each *requirement* is an *expression* that may (or may not) be able to be generated by the *CNL* – this is indicated as 0..1 multiplicity [12].

The expressiveness of a CNL is *some measure* of the variety of lexical and grammatical constructions it allows [4]. In our previous work, we proposed two measures of expressiveness as follows [12]:

- Syntactic Expressiveness is the size of the set of unique ‘E’ *expressions* that can be generated from the CNL.
- Semantic Expressiveness is the size of the set of unique ‘M’ *meanings* that can be generated from the CNL.

To achieve our design goal of *sufficient expressiveness* means that when removing ‘L’ words from the lexicon we must ensure that the ‘M’ meanings that are relevant to the ‘R’ requirements from our domain of interest are preserved, i.e. the intention is only to remove redundant and irrelevant words.

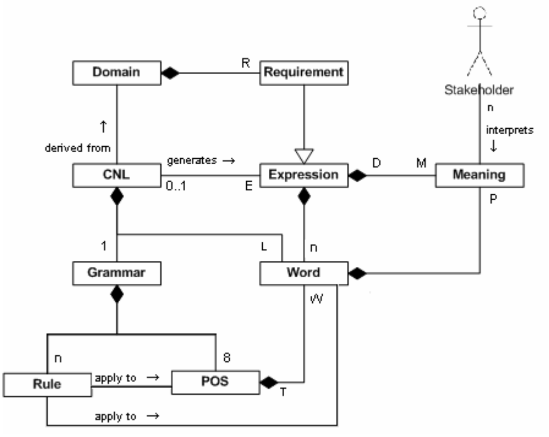

Fig. 1. CNL Abstract Model

2.3 Design Goal #3: To Be Unambiguous

The IEEE Recommended Practice for Software Requirements Specifications [14] states that "An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation." This definition is consistent with that of Kamsties [15], Davis [16] and Harwell [17]. According to Gause and Weinberg [18], ambiguity has two sources, missing information and communication errors. Missing information has various reasons. For instance, humans make errors in observation and recall, tend to leave out self-evident and other facts, and generalize incorrectly. Communication errors occur because of expression inadequacies in the writing.

There is a relationship between the expressiveness of a language, and the number of communication errors that result from the use of the language. Typically, the more constrained the lexicon, the more polysemous each word needs to be to maintain semantic expressiveness. Kamprath [7] believes that reducing polysemy is one way of reducing communication errors, since constraining each lexical term to a single meaning prevents miscommunication of the word sense. The corollary to this is of course an increase in lexical magnitude.

We believe that there are certain parts of speech that encourage authors to "leave out self-evident and other facts, and to generalize incorrectly". The parts of speech we are referring to are adjectives and adverbs. It is commonly felt that restricting the use of adjectives and adverbs should be a goal of any CNL: authors should be forced into using proper nouns (rather than adjectives and common nouns) and articulating performance requirements unambiguously (rather than using adverbs).

3 Optimal-Constraint Process – Description

3.1 Introducing Replaceability

In this section we discuss the concept of *replaceability* and propose a measure that can be used to optimise the constraining process with respect to our chosen Design

Goals. Before defining replaceability, it is important to understand the underlying concept of *similarity*. Measures of similarity quantify how much two meanings are alike and are therefore useful in identifying redundancy in a language. Similarity is a well-defined subset of relatedness which includes synonyms, hypernyms and hyponyms/troponyms [21].

Whilst Miller and Charles [22] claim that similarity tools provide some measure as to the degree of contextual interchangeability, or the degree to which one word can be substituted by another in context, they can be misleading if used carelessly. For example, there is a path length of four between 'apple#n#1'[1] and 'orange#n#1' in WordNet (where path length is defined as the number of concepts between the two terms), and while an apple and orange are similar in that they are both edible fruit – it would be misguided to think that either term could replace the other in a CNL.

There is also the issue that whilst all 'apples' are 'edible fruit', not all 'edible fruit' are 'apples'. In other words, whilst you may be able to replace a specific concept with a more general concept (i.e. hypernym) – you should not replace a general concept with a more specific concept (i.e. hyponym). This presents an ontological dimension to the CNL design. The question here is how the relative positioning of a concept within the semantic network affects its ability to be replaced by another (similar) concept. Whilst "similarity" is a specialised form of "relatedness" [21], we propose that a new concept "replaceability" be introduced that represents a specialised form of "similarity".

Replaceability: We define replaceability(x,y) as a measure of the ability of a concept 'x' to be replaced by another concept 'y' given a particular domain. Replaceability is asymmetric because there is no guarantee that the inverse replacement will be valid. This is particularly the case where a concept has been replaced by its hypernym (for instance, not all 'edible fruit' are 'apples'). We believe that "replaceability" should be a function of similarity, conventionality, polysemy, and lexical ontology. We propose the following measure:

$$Replaceability\ (x,y) = Similarity\ (x,y) \cdot \frac{F_y}{F_x} \cdot \frac{P_x}{P_y} \tag{1}$$

Where:

1. *F_x/P_x is the frequency/polysemy of x within the requirements sample, and*
2. *'y' is a synonym (or hypernym) of 'x', and*
3. *Similarity(x,y) ≥ Similarity Threshold, and*
4. *Similarity is a unity-normalised measure.*

Then:

Replacebility(x,y) ≥ 1 means x can be replaced by y.
Replacebility(x,y) < 1 means x cannot be replaced by y.

This proposed measure for replaceability addresses our three design goals. Readability is addressed since replaceability(x,y) is increased when 'y' is used more frequently in the domain than 'x'. Communications ambiguity is addressed since replaceability(x,y) is increased when 'y' is less polysemous than 'x', and ambiguity relating to "incorrect generalisation" is addressed by considering the lexical ontology and limiting replacements to synonyms and similar hypernyms only. Semantic

[1] We use the notation *word#pos#sense* to unambiguously define the meaning of *word. apple#n#1* refers to the first sense of the noun *apple* in WordNet.

Expressiveness is addressed since a word will only be replaced if there is another word that is a synonym or a (similar) hypernym, and that is used more frequently and/or less polysemously. If there is not a word that meets this criteria, then the original word is retained in the CNL. I.e. if a words meaning cannot be semantically expressed by another lexical term, then the original lexical term is retained.

Despite the replaceability rule whereby a word can only be replaced by its synonym or similar hypernym, there remains a potential for "incorrect generalizations" resulting in an increase in ambiguity. A good example of this might be if our requirements Reqt_{NL} sample was extracted from the specification for the Control Computer within an Automatic External Defibrillator (AED) – a piece of medical equipment used in the defibrillation of the heart. Within this specification, the verbs "reboot#v#1 -- *cause to load an operating system and start the initial processes*" and "resuscitate#v#1 -- *cause to regain consciousness*" would probably be encountered. Counting the nodes between these two verbs in WordNet [23] we get a path length of two (reboot#v#1 ⇔ resuscitate#v#1), which means that the concepts are very similar. Given our proposed measure for replaceability, there is great potential for resuscitate#v#1 to become the CNL term to replace reboot#v#1 (given the hypernymic relationship). It would (of course) be totally unconventional to ever replace the verb reboot#v#1 with its hypernym resuscitate#v#1. If this "incorrect generalisation" was permitted to occur, then the CNL may well increase ambiguity (rather than achieving its goal to be unambiguous).

In the context of our AED example, the verb reboot#v#1 would likely be used when talking about the control computer, and the verb resuscitate#v#1 would likely be used when talking about the human patient. Interestingly, the shortest path between the object nouns computer#n#1 and human_being#n#1 in WordNet is quite long (at a length of 16) [23]. So although the two verbs are very similar, the fact that their object nouns are so dissimilar may provide the extra dimension of information that is required to prevent this "incorrect generalisation". So far we have not discussed the scope of words (i.e. 'x' and 'y') that are measured against each other for replaceability. For example, is it possible that we could use our knowledge of dissimilarity between the object nouns computer#n#1 and human_being#n#1 to prevent the comparison of reboot#v#1 and resuscitate#v#1 (such that reboot#v#1 does not get replaced by resuscitate#v#1)?

The replaceability measure that we presented above will work for any scope of words and is not sensitive to inter-relationships between parts of speech. We propose that instead of modifying the replaceability measure to account for inter-relationships between parts of speech, we introduce the concept of Replaceability Matrices to manage the scope of words that are *appropriate* to be compared to each other for replaceability. By *appropriate,* we mean that the words within a single Replaceability Matrix are all from the same part of speech, and all associate with *similar* words from grammatically related parts of speech (we discuss this in more detail (for verbs) in Section 4.1.1).

3.1.1 Replaceability Matrices

The Replaceability Matrix in Table 1 is effectively an N^2 matrix that we will use to capture replaceability measurements for words from the same part of speech that are associated with *similar* words from related parts of speech. We will use the Replaceability Matrix to constrain the lexicon, since we will be making decisions on which words are to be replaced.

Table 1. Replaceability Matrix

NL	$X_{NL}(F_X)(P_X)$	$Y_{NL}(F_Y)(P_Y)$	$Z_{NL}(F_Z)(P_Z)$
$X_{NL}(F_X)(P_X)$	$Repl(X_{NL},X_{NL})$	$Repl(X_{NL},Y_{NL})$	$Repl(X_{NL},Z_{NL})$
$Y_{NL}(F_Y)(P_Y)$	$Repl(Y_{NL},X_{NL})$	$Repl(Y_{NL},Y_{NL})$	$Repl(Y_{NL},Z_{NL})$
$Z_{NL}(F_Z)(P_Z)$	$Repl(Z_{NL},X_{NL})$	$Repl(Z_{NL},Y_{NL})$	$Repl(Z_{NL},Z_{NL})$

To understand Table 1, it is essential to understand that X_{NL} is a concept that is comprised of a NL lexical term X as well as a PoS and a sense (resulting from the shallow parsing and Word Sense Disambiguation (WSD) respectively). Note that (F_X) means the Frequency of X_{NL} as relevant to this Replaceability Matrix. Therefore, if X_{NL} happens to be a verb that is also used with other dissimilar subject (or object) nouns, then it would have other F_X's as applicable to each of the other Replaceability Matrices. Similarly, (P_X) means Polysemy of X_{NL} with respect to this specific Replaceability Matrix, i.e. (P_X) does not mean the polysemy of X_{NL} as found in a dictionary. Using a dictionary will likely over-inflate the polysemy count of many words that may be unambiguously used within the domain. Y_{NL} and Z_{NL} have been used in Table 1 to give the impression that typically there will be a number of concepts being compared in a Replaceability Matrix. Each intersecting cell in the Replaceability Matrix represents the Replaceability between two concepts, i.e. $Repl(X_{NL},Y_{NL})$ measures the ability of concept X_{NL} to be replaced by concept Y_{NL}. The following rule applies.

Rule #1: The concept at the start of a row is replaced by the concept corresponding to the column having the highest replaceability value on that same row.

Notice that it is possible for a concept to be selected as the replacement for itself – which in effect means the original NL term is retained. This is exactly how the CNL achieves its goal of being sufficiently expressive.

3.2 Optimal-Constraint Process

Figure 2 presents the process that we have developed to optimally constrain the lexicon of a CNL. The process is optimised in the sense that we employ a replaceability measure that is focused on achieving our design goals. One of the major challenges with optimally constraining a lexicon is determining which words are redundant or irrelevant and can be removed without reducing the semantic expressiveness of the language for a selected domain of interest. One of the novel contributions of this research is the application of existing NLP tools and techniques to this process, such that the result is goal-optimised and repeatable.

Figure 2 shows that the design process begins with a NL requirement ($Reqt_{NL}$). The first step is to shallow parse the $Reqt_{NL}$ to determine the parts of speech and grammatical phrases. Shallow Parsing can be used to perform tokenisation, POS tagging, and phrase boundary detection (e.g. noun phrases, verb phrases, prepositional phrases, etc.) such that grammatical relations can be identified [19]. Word Sense Disambiguation (WSD) would then occur aiming to associate a given word in a passage of text with the authors original intended meaning or sense [20]. At this point, each word in each $Reqt_{NL}$ could be represented in the form of word#pos#sense.

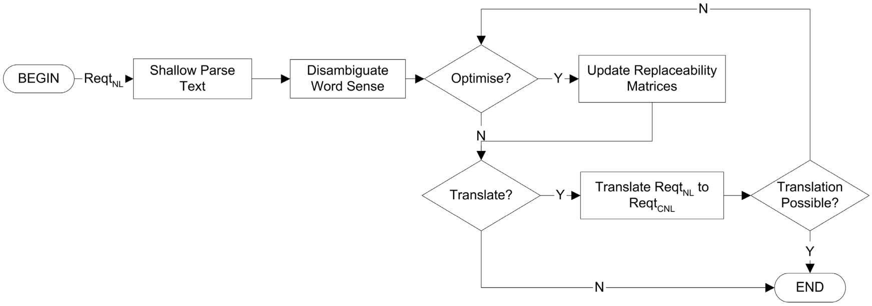

Fig. 2. Optimal-Constraint Process Flowchart

The ‘Optimise?’ and ‘Translate?’ decision points reflect two fundamentally different phases in the life of a constrained lexicon. The first phase could be considered the ‘setup phase’, where the Reqt_{NL} sample would be injected into the optimal-constraining process to update the Replaceability Matrices (recall Section 3.1.1). The second phase could be considered the ‘operating phase’ where the established Replaceability Matrices are then used to replace each NL Requirement (Reqt_{NL}) with its semantically equivalent CNL alternative (Reqt_{CNL}). In ‘setup phase’ the constraining process would typically be optimising but not translating. In ‘operating phase’ the constraining process would typically be translating but not optimising. The ‘Translation Possible?’ decision allows for the event whereby Reqt_{NL} contains terms that have no CNL translation in the established Replaceability Matrices. In this case it is possible to optimise the Replaceability Matrices to accommodate the new concept – ensuring that the constrained lexicon maintains sufficient expressiveness. Ideally, inexperienced authors would be prevented from optimising the CNL such that is does not accommodate their ‘bad habits’.

4 Optimal-Constraint Process – Design Decisions and Rationale

Whilst the Optimal-Constraint Process Description (Section 3) is intended to be non-implementation specific and thus future-proof, the design decisions presented in this section are based upon the capability of currently available technology. The expectation is that as NLP technology improves, future researchers can revise these decisions without needing to revisit the Optimal-Constraint Process Description.

4.1 Parts of Speech to Constrain

Whilst it may be theoretically possible to apply the optimal-constraint process to each of the eight main parts of speech, there are two reasons why we currently limit the application of our process to verbs.

Firstly, subject and object nouns in requirement text are often domain-specific proper nouns (e.g. the “SPS-49 Air Search Radar” rather than the “long-range high-power radar”). The use of proper nouns also means that adjectives are rarely used in requirement text (in fact, experts often recommend against the use of adjectives and adverbs as they are seen as vague words [24]). Function words (determiners, prepositions, conjunctions, and pronouns) are already closed parts of speech and it could be argued that further constraining the lexicon in these parts of speech is unnecessary. Interjections are, by their nature, inappropriate for use in technical writing [13]. Therefore, when constraining the lexicon for writing requirements, it could be argued that verbs are the only part of speech that should be constrained in this way. Given there are over 29,000 verbs in the English language [25] and that on average, verbs are the most polysemous part of speech [23], constraining verbs seems to be necessary.

Secondly, the semantic networks that are in existence today do not manage hypernymic or hyponymic (/troponymic) relationships between these other parts of speech. Presently they are limited to nouns and verbs only. Miller [26] states that updating WordNet with *is-a* relationships for adjectives and adverbs is a work in progress.

4.1.1 Scope of Replaceability Matrices for Verbs

In Section 3.1 we introduced the concept of the Replaceability Matrix to be used as the mechanism to manage the scope of words from the same part of speech, that are associated with *similar* words from grammatically related parts of speech. For verbs, the grammatically related parts of speech would be the subject noun and object noun (transitive verbs). The following rule is proposed.

Rule #2: If

Verb$_A$#Verb relates to Subject$_A$#Noun and Object$_A$#Noun, and

Verb$_B$#Verb relates to Subject$_B$#Noun and Object$_B$#Noun;

then, Verb$_A$ and Verb$_B$ can only exist in the same Replaceability Matrix if Subject$_A$ and Subject$_B$ are *similar* AND if Object$_A$ and Object$_B$ are *similar*.

4.2 Shallow Parsing and Word Sense Disambiguation

We decided to use the Memory Based Shallow Parser (MBSP) [27] to identify phrase chunks in simple sentence requirements. Daelemans [27] claims that the MBSP is over 90% accurate for noun and verb phrase detection, making the MBSP one of the more accurate shallow parsers available. Manning [19] suggests that whilst NLP taggers and chunkers can mine data automatically, it is often the case that in order to obtain accurate results, the process must be highly interactive. We therefore decided to use a human inspection to confirm the results of the MBSP.

Although there are automated WSD tools freely available, we trialled both WordNet::SenseRelate [28] and Sense Learner 2.0 [29] with both tools failing to accurately disambiguate the requirement text in the majority of cases. The disappointing results are believed to stem from the fact that WSD tools rely on contextual information to make a probabilistic determination on the sense of each word. For example, to disambiguate a verb – the WSD tool would look at the sense of the surrounding nouns. Given that in requirement specifications the surrounding nouns are typically domain specific proper nouns, the WSD tool was unable to make sense of the necessary contextual information. Interestingly, Manning finds that human performance is typically the upper bound for WSD [19]. For this reason we decided to manually disambiguate the sense of each word. We used WordNet [23] as the reference dictionary.

4.3 Similarity Measurement

4.3.1 Similarity Measure

The decision has been made to use WordNet [23] as the semantic network for defining and relating lexical concepts. WordNet is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, and adjectives are organized into synonym sets, each representing one underlying lexical concept [26]. Synonym sets are then associated with other synonym sets via lexical relationships (e.g. synonymy, antonymy, hyponomy ("is a"), meronymy ("part of"), and morphological relationships). WordNet::Similarity [21] is a tool that draws upon the lexical network of WordNet to provide a measure of similarity between any two words from the same Part of Speech. There are three inputs required for this tool to operate: word1#pos#sense, word2#pos#sense, and the chosen Similarity Measure. The output is a value representing the similarity between the two concepts. We decided to use the Wu and Palmer [30] similarity measure since its developers described this measure to be most appropriate to a verb taxonomy.

4.3.2 Similarity Threshold

The Similarity Threshold is perhaps the most instrumental factor in trading off readability, expressiveness, ambiguity, and lexical magnitude. In general, the higher the Similarity Threshold the better the readability and expressiveness since there will be fewer lexical replacements (and therefore more of the original and conventional NL words will be available within the CNL lexicon). Ambiguity relating to "missing information" will likely be reduced with a higher similarity threshold, since there will be a reduced potential for "incorrect generalizations". On the other hand, it may be possible to worsen the ambiguity relating to "communications errors" by raising the Similarity Threshold, since words may be prevented from being replaced by less polysemous, or more conventional alternatives.

When the Similarity Threshold is increased, so too is the number of Replaceability Matrices, since there will be reduced similarity between Subjects and between Objects. Additionally, within each of the Replaceability Matrices, there will be reduced similarity between verbs – resulting in reduced lexical replacements (and therefore less reduction in the CNL lexical magnitude). In summary, the disadvantages of having a high Similarity Threshold are that the resulting CNL lexical may be large, and communications ambiguity may not be reduced by allowing less polysemous, or more conventional replacements. Whilst we cannot recommend one magical similarity threshold value that will work in all situations, we have found through our own empirical research [12] that a similarity threshold of 0.6-0.7 seems to achieve a reasonable trade-off between syntactic expressiveness and lexical magnitude when using WordNet::Similarity with the Wu & Palmer measure.

5 Applying the Process – Example

The following example aims to solidify the readers understanding of our process by applying it to a small sample of hypothetical requirements. Table 2 includes three columns. 'ID' is an arbitrary requirement identifier, 'Reqt$_{NL}$' and 'Reqt$_{CNL}$' present the sample requirements before and after replacement respectively. We limit our example to the constraining of verbs as per the decision made in Section 4.1.

Table 2. Example Requirements – Reqt$_{NL}$ and Reqt$_{CNL}$

ID	Reqt$_{NL}$	Reqt$_{CNL}$
Req-01	The radar shall track aeroplanes...	The radar shall ~~track~~observe aeroplanes...
Req-02	The radar shall monitor helicopters...	The radar shall ~~monitor~~observe helicopters...
Req-03	The radar shall observe aircraft...	The radar shall observe aircraft...
Req-04	The 3d radar shall observe missiles...	The 3d radar shall observe missiles...
Req-05	The radar shall monitor the interface...	The radar shall monitor the interface...
Req-06	The captain shall be able to watch helicopters...	The captain shall be able to watch helicopters...
Req-07	The radar shall watch meteorological balloons...	The radar shall watch meteorological balloons...

Notice that Req-01-Req-07 have been simplified by truncating the Prepositional Phrases (PP) that follow the Subject-Verb-Object triple. This is because our process does not rely upon PP information to constrain verbs. For instance Req-06 should probably state "The captain shall be able to watch helicopters *from standing on the bridge*". The first step of the process is to shallow parse the Reqt$_{NL}$ text. Using the Memory Based Shallow Parser [27] on Req-01 gives:

[NP$_1^{\text{Subject}}$ The/DT radar/NNP NP$_1^{\text{Subject}}$] [VP$_1$ shall/MD track/VB VP$_1$] [NP$_1^{\text{Object}}$ aeroplanes/NNP NP$_1^{\text{Object}}$]

Manually using WordNet, we can then disambiguate the sense of the subject "radar" as *"measuring instrument in which the echo of a pulse of microwave radiation is used to detect and locate distant objects"* which is a noun with sense #1. We represent this in shorthand as radar#n#1. Similarly we can do this for the verb "track" and object "aeroplane" to get track#v#2 and aeroplane#n#1 respectively. We could then continue this process for Req-02 to Req-07.

Figure 3 illustrates the result of applying Rule #2 on our sample requirements. The "Verb" section in Figure 3 shows how we would determine the number and composition of each Replaceability Matrix based on identifying similar words from grammatically related parts of speech, i.e. for verbs there is the relationship to similar subjects and similar objects (the ovals illustrate the groupings of similar concepts).[2]

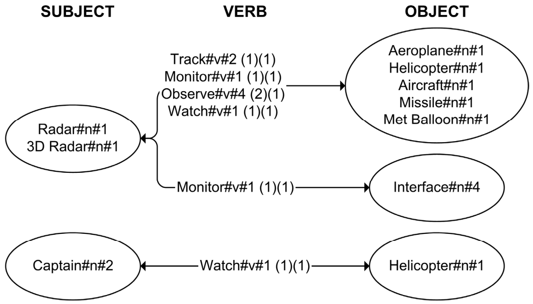

Fig. 3. Subject-Verb-Object Relationships

[2] Note that throughout this example, we use WordNet::Similarity and the Wu & Palmer similarity measure with a Similarity Threshold of 0.6.

Table 3 shows the first of the three Replaceability Matrices. As an example, consider track#v#2 and observe#v#4 as our ‘x’ and ‘y’ respectively in the replaceability measure. Note that observe#v#4 is a hypernym of track#v#2[3] in WordNet [23].

Table 3. Replaceability Matrix – Example

VERB					CNL
	track#v#2 (1)(1)	observe#v#4 (2)(1)	monitor#v#1 (1)(1)	watch#v#1 (1)(1)	
track#v#2 (1)(1)	1	1.72	Not Hyp/Syn	0.67	observe#v#4
observe#v#4 (2)(1)	Not Hyp/Syn	1	Not Hyp/Syn	0.4	observe#v#4
monitor#v#1 (1)(1)	Not Hyp/Syn	1.72	1	0.67	observe#v#4
watch#v#1 (1)(1)	Not Hyp/Syn	Not Hyp/Syn	Not Hyp/Syn	1	watch#v#1

Given that $F_X = 1$, $P_X = 1$, $F_Y = 2$, $P_Y = 1$, we get a replaceability measure of 1.72. Rule #1 states that the concept at the start of the row (track#v#2) *is replaced by the concept corresponding to the column having the highest replaceability value on that same row* (observe#v#4), so the replacement for track#v#2 is observe#v#4. This same process would be applied to all rows in the three Replaceability Matrices. Note that we have not shown the Replaceability Matrices for Monitor#v#1 or Watch#v#1 since these would only contain a single verb, and would end up being replaced by themselves – resulting in no constraining of the lexicon. The end result of this example can be seen in Table 2 where we have re-written the requirements using the constrained lexicon (Reqt_{CNL}).

Some key observations from Table 2 Reqt_{CNL} column: Notice that “observe” seems to be a reasonable replacement for the verbs “track” and “monitor” in Req-01 and Req-02 respectively. Notice that “monitor” in Req-05 is not replaced since it was part of a different Replaceability Matrix (recall Figure 3). Notice that “watch” cannot be replaced in Req-07 since the Replaceability Matrix contains no other words which are hypernyms or synonyms (i.e. watch#v#1 is more general than the other terms).

6 Limitations and Future Work

6.1 Replaceability Measure

Our proposed measure for replaceability is somewhat simplistic in that it does not put weightings on the relative importance of similarity vs. frequency vs. polysemy. For instance, when considering the replaceability(x,y), this means that a ‘y’ with half the polysemy count is equally as replaceable as a ‘y’ that is used twice as frequently. One improvement would be to introduce weightings, whereby we could weight the relative importance of similarity vs. frequency vs. polysemy. Furthermore we could even use requirement weightings to put some weighting on the importance of each lexical term.

Another limitation with our proposed replaceability measure was to restrict lexical replacements to synonyms and hypernyms only in an attempt to prevent “incorrect generalisations” (you can’t compare apples with oranges!). In some cases, this may prove to be overly conservative, resulting in an under-constrained CNL. There are

[3] The similarity between the two terms is 0.86.

possibly situations where it would be appropriate to replace a word with its coordinate (sibling) term. For example, consider the coordinate verbs save#v#2, store#v#1, and retain#v#3. It could be argued that a lexicon may not be optimally constrained if it were to retain all three of these terms.

6.2 Integrating with Constrained Grammars

The ideas presented in this paper on constraining a lexicon complement current research on constraining grammars. For example, ACE [2], PENG [3], and Grover [6] all have a constrained grammar and a constrained lexicon of function words (determiners, prepositions, conjunctions, and pronouns), but allow the user to invent their own list of content words (verbs, nouns, adjectives, adverbs). The problem with this is that there is no guidance given to the user as to how they might go about deriving such a list (e.g. how would they decide which verbs to include?). This of course is the very focus of our paper. Given that our process is specifically targeted at deriving such "content words" from the domain of interest, we strongly believe that the two branches of research are complementary (and non-overlapping). Combining these two areas of research may empower the analyst to do consistency checking and logical reasoning (for example they could query the resultant specification for all of the inputs and outputs of a specified subject noun by looking for verbs *similar to* "accept" and "provide" respectively).

7 Conclusion

The aim of this paper was to present a fully automatable NLP-based process for optimally constraining the lexicon of a CNL. Our optimal-constraint process is significant since we have identified a new application for existing NLP tools and techniques that ensures a rigorous and repeatable outcome, and means we potentially no longer require a lexicographer to manually sift through the large volume of text and make (possibly unrepeatable and unjustifiable) subjective decisions on the content of the lexicon. We bounded 'optimal-constraint' by defining three design goals for the constrained lexicon, to be *readable*, *sufficiently expressive*, and *unambiguous*. We proposed a new concept 'replaceability', which we argued provides a better measure than 'similarity' as to the degree of contextual interchangeability, or the degree to which one word can be substituted by another in context. This is because 'replaceability' is a function of conventionality (frequency), polysemy, lexical ontology and similarity – rather than similarity alone, which we argue can be misleading.

Although not a limitation of the process, we did find that the immaturity of WSD tools prevented total automation of the process. This limitation is considered to be time-sensitive, and reflective of the current (developmental) state of NLP technology. It is expected that as WSD algorithms and tools improve this limitation will cease to exist, and complete automation will be possible. Our process theoretically makes it possible to automatically generate a constrained lexicon from a sample of requirements. We believe that our process is pragmatic and accessible since it relies on nothing more than existing NL requirement specifications, freely available NLP tools, and domain knowledgeable individuals.

The next stage in our longitudinal study will be to empirically validate that our optimal-constraint process actually does achieve its design goals by using a domain specific requirements sample. The resultant lexicon will then be the subject of a controlled experiment to measure the effects on the respective quality factors (readability, expressiveness, and ambiguity).

References

1. The Standish Group International CHAOS Report 1994, The Standish Group International, Inc., Massachusetts http://www.standishgroup.com/sample_research/chaos_1994_1.php
2. Fuchs, N.E., Schwitter, R.: Attempto Controlled English (ACE). In: Proceedings of the First International Workshop on Controlled Language Applications, Belgium, pp. 124-136 (1996)
3. Schwitter, R., English, R.: as a formal specification language. In: Proceedings. 13th International Workshop on Database and Expert Systems Applications, Aix-en-Provence, pp. 228–232 (2002)
4. Somers, H. (ed.): Computers and Translation: A Translator's Guide, John Benjamins Publishing Company, Amsterdam (2003)
5. Goyvaerts, P.: Controlled English, Curse or Blessing? - A User's Perspective. In: Proceedings of the First International Workshop on Controlled Language Applications, Belgium (1996)
6. Grover, C., Holt, A., Klein, M. M.: Designing a Controlled Language for Interactive Model Checking. In: Proceedings of the Third International Workshop on Controlled Language Applications, Washington, pp. 90–104 (2000)
7. Kamprath, C., et al.: Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English. In: Proceedings of the Second International Workshop on Controlled Language Applications, Pennsylvania, pp. 51–61 (1998)
8. AECMA1986, AECMA/AIA Simplified English: A Guide for the Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintenance Language, Association Europeenne des Constructueurs de Materiel Aerospatial (1986)
9. Swaffar, J.: What makes text readable?, University of Texas, Austin, http://www.utexas.edu/ courses/swaffar/distance/review.html
10. Gnesi, S., et al.: An Automatic Tool for the Analysis of Natural Language Requirements. International Journal of Computer Systems Science & Engineering 20(1), 53–62 (2005)
11. Fabbrini, F., Fusani, M., Gervasi, V., Gnesi, S., Ruggieri, S.: Achieving Quality in Natural Language Requirements, 11th International Software Quality Week, San Francisco (1998)
12. Boyd, S., Zowghi, D., Farroukh, A.: Measuring the Expressiveness of a Constrained Natural Language: An Empirical Study. In: Proceedings of the 13th International Conference on Requirements Engineering, Paris (2005)
13. Quirk, R., Greenbaum, S.: University Grammar of English. Longman, London (1996)
14. IEEE1993, IEEE Recommended Practice for Software Requirements Specifications, ANSI/IEEE Standard 830-1993, New York (1993)
15. Kamsties, E., Berry, D.M., Paech, B.: Detecting Ambiguities in Requirements Documents Using Inspections, Workshop on Inspections in Software Engineering (WISE'01), Paris (2001) pp. 68–80 (2001)
16. Davis, A., et al.: Identifying and Measuring Quality in a Software Requirements Specification, First International Software Metrics Symposium, Baltimore, pp. 141–152 (1993)

17. Harwell, R., Aslaksen, E., Hooks, I., Mengot, R., Ptack, K.: What is a Requirement?, Proceedings of the Third Annual International Symposium, National Council of Systems Engineers (NCOSE), pp. 17–24 (1993)
18. Gause, D.C., Weinberg, G.M.: Exploring Requirements: Quality Before Design, Dorset House, New York (1989)
19. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
20. Ide, N.V, éronis, J.: Word Sense Disambiguation: The State of the Art. Journal of Computational Linguistics 24(1), 1–40 (1998)
21. Pedersen, T., et al.: WordNet:Similarity - Measuring the Relatedness of Concepts, Nineteenth National Conference on Artificial Intelligence, San Jose (2004)
22. Miller, G.A., Charles, W.G.: Contextual Correlates of Semantic Similarity. Language and Cognitive Processes 6(1), 1–28 (1998)
23. Web WordNet 2.0, Cognitive Science Laboratory Princeton University, Princeton (2003), http://wordnet.princeton.edu/cgi-bin/webwn
24. Fabbrini, F., Fusani, M., Gnesi, G., Lami, G.: An Automatic Quality Evaluation for Natural Language Requirements. In: Proceedings of the Seventh International Workshop on RE: Foundation for Software Quality, Interlaken, Switzerland (2001)
25. Oxford English Dictionary, Oxford University Press, New York (2006)
26. Miller, G.A., et al.: Five papers on WordNet, Special Issue of International Journal of Lexicography, vol. 3(4) (1990)
27. Daelemans, W., Buchholz, S., Veenstra, J.: Memory-based Shallow Parsing. In: Proceedings of CoNLL-99, Bergen (1999)
28. Patwardhan, S., Banerjee, S., Pederson, T.: SenseRelate: TargetWord - A Generalized Framework for Word Sense Disambiguation, Twentieth National Conference on Artificial Intelligence (Intelligent Systems Demonstration), Pittsburgh (2005)
29. Mihalcea, R., Faruque, E.: SenseLearner: Minimally Supervised Word Sense Disambiguation for All Words in Open Text, Proceedings of ACL/SIGLEX Senseval-3, Barcelona (2004)
30. Wu, Z., Palmer, M.: Verb Semantics and Lexical Selection, 32nd Annual Meeting of the Association for Computational Linguistics. Las Cruces, New Mexico (1994)