

Speech Detection of Stakeholders' Non-Functional Requirements

Adam Steele, Jason Arnold, Jane Cleland-Huang

Center for Requirements Engineering

School of Computer Science, Telecommunications and Information Systems

DePaul University

{asteel,jarnold,jhuang}@cs.depaul.edu

Abstract

This paper describes an automatic speech recognition technique for capturing the non-functional requirements spoken by stakeholders at open meetings and interviews during the requirements elicitation process. As statements related to system qualities such as security, performance, and portability are often scattered throughout statements of functional need, the ability to “listen in” on a conversation and correctly capture these statements into a single view is very helpful. The approach is intended to enhance and not replace existing elicitation methods in which stakeholders are more directly asked to describe their needs. Training a speech detection tool to recognize individual users is time consuming while speech detection for un-enrolled users is notoriously difficult. Our approach uses a context-free grammar to boost recognition accuracy, segment the stakeholders’ utterances and finally to classify the recognized statements by quality type. This paper describes the preliminary results from experiments with different subjects and then discusses methods for optimizing the recognition and capture of non-functional requirements and contextual domain terms.

1. Introduction

During the requirements elicitation process stakeholders tend to focus upon the functionality of the system rather than on its non-functional requirements (NFRs) [1,2]. This can often result in failure to consider critical attributes of the system until late in the development process, at which time these attributes may become more difficult and more expensive to fully implement [3]. In prior work we described a technique for automating the detection and classification of NFRs and early aspects from a broad range of documents such as interview notes, unstructured requirements and memos [4]. Early identification of a stakeholder’s non-functional needs, enables potential conflicts and tradeoffs to be more fully considered during the early analysis and design [3]. This in turn can lead to an improved design that more

fully meets the real needs of the stakeholders. This paper builds on our prior work in NFR detection, by addressing the challenging problem of detecting spoken NFRs captured during meetings with stakeholders. The benefits of a speech-based NFR detection tool are primarily in its ability to identify stakeholder needs that may not have otherwise been considered. It is most effective when used to trawl through discussions of large numbers of stakeholders in order to identify potential NFRs that can then be further explored through more specific interviews.

Although speech recognition software has advanced considerably over the past decade, it is still most effective when trained (*enrolled*) for an individual’s voice [5]. Unfortunately in many situations, *speaker dependent* recognition is impractical due to the time needed to train the system for individual speakers and also because there are too many people to enroll. Without enrollment, recognition accuracy is often low. To illustrate this point, consider the following sentence representing the stakeholder statement: “We want a conservative look and feel.” Without prior enrollment of the speaker, this sentence was recognized as “We want a conservative block conceal.” Similarly the sentence “Payroll must take less than thirty minutes to generate”, was recognized as “Payroll mistake lesson thirty minutes degenerating.”

Fortunately, most automatic speech recognition (ASR) tools allow the definition of context free grammars which can constrain the recognition of the utterances, and boost recognition accuracy. Generally, such context free grammars have been used to capture speech in predictable contexts. For example, a grammar could be constructed to interpret the responses of a customer placing an order for an item over the phone, to check airplane departure times, or accept commands in other similarly predictable interactions. Capturing and deriving meaning from free speech in a loosely constrained environment represents a significantly more challenging problem.

This paper therefore introduces an initial proof of concept approach for tackling this problem. Section two briefly summarizes our prior work on the detection and classification of NFRs and aspects from written documents. Section three discusses the technical challenges of speech recognition and word spotting.

Table 1. Top ten stemmed terms ranked by indicator strength for selected NFR types

NFR type	Top ten terms ranked by indicator strength
Availability	Avail, achiev, hour, time, dai, year, pm, technic, system, long
Legal	Regul, compli, standard, sarban, oxlei, php, pear, legal, estima, law
Security	Onli, access, author, user, inform, data, ensur, system, authent, secure
Usability	Us, easi, user, train, product, abl, successfully, understand, system, learn
Performance	Second, respons, time, longer, fast, minut, system, take, user, process

Section four then introduces the proposed approach for speech based NFR recognition and discusses its underlying grammatical model. Section five describes a simple experiment we conducted to evaluate the ability of the algorithm to correctly detect and classify NFR related concerns spoken by multiple stakeholders. The paper then concludes with a discussion of potential applications and limitations of this approach within the requirements elicitation process.

2. Detecting Non-Functional Requirements

In prior work, Cleland-Huang et al described a method for the automated detection of NFRs and aspects from written text [4]. Training sets of pre-categorized requirements were used to discover terms that would serve as indicators for a given quality type. During the training process, all words were stemmed to their root forms, so that different derivatives of the same word would be considered equivalent. The strength of a given indicator term t in respect to a specific quality type Q was calculated as follows:

$$\Pr_Q(t) = \frac{1}{N_Q} \sum_{i=1}^{N_Q} \frac{\text{freq}(d_{Q,i}, t)}{|d_{Q,i}|} \cdot \frac{N_Q(t)}{N(t)} \cdot \frac{NP_Q(t)}{NP_Q} \quad (1)$$

A complete explanation of this expression is given in [4].

The first term $\frac{1}{N_Q} \sum_{i=1}^{N_Q} \frac{\text{freq}(d_{Q,i}, t)}{|d_{Q,i}|}$ computes the average

frequency of term t in requirements of type Q normalized by the size of the document $|d_{Q,i}|$. The second term $\frac{N_Q(t)}{N(t)}$

measures the number of requirements of type Q that contain t in respect to the total number of requirements that contain t . This expression computes to 1 if all occurrences of term t are found in requirements of type Q .

The final factor, $\frac{NP_Q(t)}{NP_Q}$ computes the number of projects

that contain the term t in type Q requirements over the total number of projects containing type Q requirements. This factor weights terms that appear in type Q NFRs across multiple projects more highly than those that appear in fewer projects. Table 1 shows the top ten

indicator terms ranked by strength for several quality types.

During the detection and classification phase, these previously mined indicator terms are used to calculate a likelihood that a given requirement or sentence belongs to a specific quality type Q . Likelihood is calculated by summing the term weights of all type Q indicator terms appearing in the sentence in respect to the total number of indicator terms appearing in the sentence. The expression is shown below.

$$\Pr_Q(R) = \frac{\sum_{t \in R \cap I_Q} \Pr_Q(t)}{\sum_{t \in I_Q} \Pr_Q(t)} \quad (2)$$

Results from an empirical study returned an average recall of approximately 75% and average precision of 26% for nine different NFR types. Several of the individual quality types, such as *availability*, *security*, *usability*, and *maintainability* returned better results than requirements related to *legal* and *look-and-feel*. The speech recognition work in this paper focuses on security, performance, and usability requirements which were retrieved at recall values of 81%, 63%, and 98% respectively, and precision values of 28%, 19%, and 28%.

3. Speech Recognition and Word Spotting

The objective of recognizing stakeholders' comments concerning NFRs during the elicitation process assumes either a meeting type environment or an individual interview. In either case, the number of stakeholders involved in the process prohibits individual user enrollment (training) which means that the recognition system must be *speaker independent*. Users are expected to interact with the system using normal, *continuous* speech. Furthermore, because of the nature of the requirements elicitation process it is infeasible to expect a highly structured dialog in which the speaker is constrained to answer a limited set of questions in rather predictable ways.

Unconstrained, continuous, and speaker independent speech recognition has always been extremely inaccurate, especially in the presence of background noise [5]. However, acceptable recognition rates have been achieved

for speaker independent systems when an appropriate command and control grammar can be defined that strongly specifies what is expected to be said at any point in the human-machine dialog [6,7]. Unfortunately, stakeholders' NFR related comments are largely unstructured, and can not be realistically constrained by such a *directed-dialog*. The challenge is therefore to impose sufficient structure to boost the recognition accuracy of the system without over-predicting what might actually be said. Over predictions lead to false assumptions about what the speaker has said and therefore reduce the system's ability to accurately recognize the types of dialog characteristic of NFR specifications.

Many researchers have investigated techniques for recognizing keywords (*word spotting*) in both discrete-word and continuous speech [8], however their models have primarily focused on relatively small vocabularies and have traditionally been optimized for telephone conversations. More recently, the Sphinx project at Carnegie-Mellon developed an open source speech recognition system that can handle larger vocabularies, and can also impose structure on the recognized speech using context-free grammars (CFGs) and recursive transition networks (RTNs) [9,10]. Nevertheless the Sphinx project focuses on speech recognition for limited tasks such as requesting bus timetables and reserving conference rooms.

For the initial experiments described in this paper we used Microsoft's SAPI 5.1 [11]. The underlying recognition engine, like most modern ASR systems is based on Hidden-Markov Models (HMM) [12] which has the advantage of associating a recognition probability with each of the recognized words. The SAPI models are speaker independent but are sensitive to background noise and over-modulation of the audio signal. Therefore the positioning of the microphone and the use of a noise-canceling microphone array, or a head-worn boom microphone is critical for recognition accuracy. Another characteristic of HMM based recognizers is that in sentences that have optional dictation components, recognition accuracy is significantly improved if the sentences are anchored by start words.

4. The NFR Speech Recognition Model

NFRs typically contain several interesting elements including indicator terms, domain terms, actions, and quantifiers. While indicator terms are used to classify requirements by NFR type, domain terms enrich the requirement by describing the context in which it is applicable. Quantifiers provide additional information about the required level of the described quality. Consider, for example, the following stakeholder statement tagged according to these four elements:

All available flights must be displayed within two seconds

domain	action	quant-	indicator
term		ifier	term

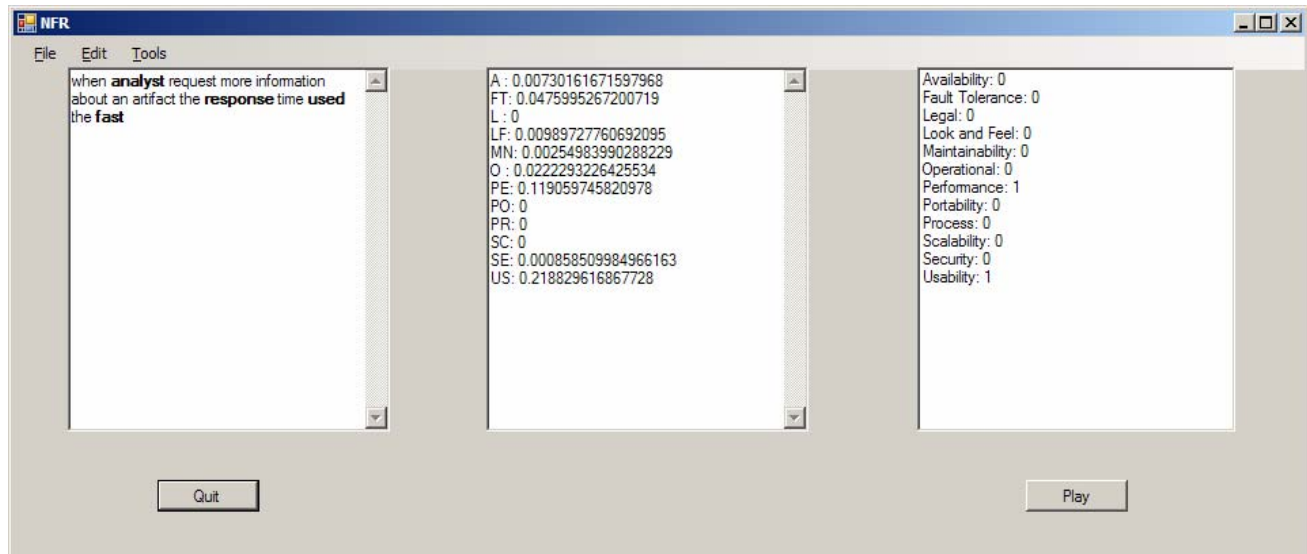
In this statement, the term "*All available flights*" provides domain information; "*displayed*" describes the action, "*seconds*" serves as an indicator for a performance requirement; and "*two*" qualifies the time requirement. The indicator terms used in this work, such as those shown in Table 1, were taken from the results of previous experiments [4]. Domain terms can be mined from early vision statements [13] and used to support the deeper analysis of the NFR. The initial work described in this paper focuses upon use of indicator terms only.

4.1 The Context Free Grammar

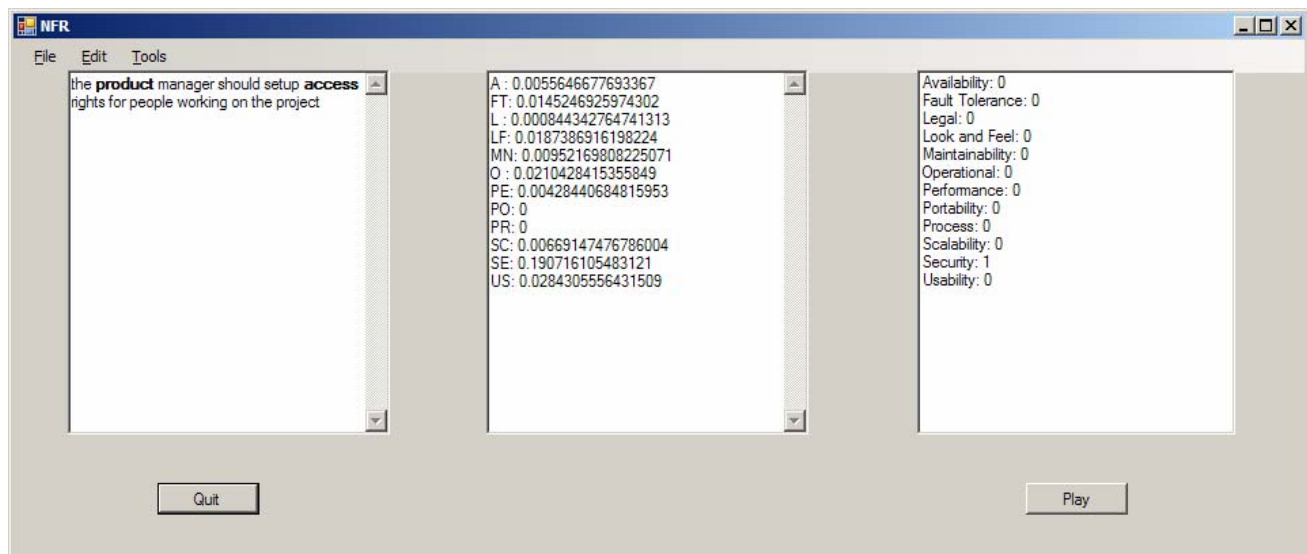
One of the powerful features of the SAPI system is that properties can be associated with each of the grammatical rules, and then extracted at runtime when the rule is actually applied. In the NFR detection grammar, property types are defined for each of the NFRs of interest. For example in the following rule-fragment, properties are defined for legal (L), look-and-feel (LF), availability (A), and maintainability (M) NFRs. In this example the term "Regul" represents an indicator term for *legal* NFRs at a weighting of approximately 0.028. The term "Avail" is a strong indicator for *availability*, and a weak indicator for *maintainability*.

```
<RULE NAME="Term">
  <L>
    ...
    <RULEREF NAME="Regulatory"
      PROPNAME="L 0.02795"/>
    <RULEREF NAME="Appear"
      PROPNAME="LF 0.01129"/>
    <RULEREF NAME="Avail"
      PROPNAME="A 0.03953,MN.00016"/>
    ...
  </L>
</RULE>
```

In the text based NFR detector, the detection phase involves parsing the target text, stemming terms to their roots, and then calculating the likelihood that a statement represents a given NFR based on the occurrence of relevant indicator terms. The problem is complicated in the speech version because the words need to be understood before they can be stemmed, and therefore the indicator terms for each of the NFRs must be expanded into all common derivative forms for inclusion in the grammar. The following rule fragment illustrates the expansion of the term "Appear" into several derivatives.



a. "When the analyst requests more information about an artifact the response time needs to be fast."



b. "The product manager should setup access rights for people working on the project"

Figure 1. NFR Speech-based detection tool

```
<RULE NAME="Appear">
  <L>
    <P>Appear</P>
    <P>Appears</P>
    <P>Appears</P>
    <P>Appeared</P>
  </L>
</RULE>
```

In our current prototype, this expansion is performed manually; however it is potentially feasible to automate this process through reversing the stemming algorithm

and checking candidate words against a thesaurus such as WordNet.

In addition to specifying individual terms and their derivatives, the grammar also specifies sentence composition rules based on various combinations of anchor words, indicator terms, and open dictation. The initial grammar used for this proof of concept study, was based on our observation of the way people tend to phrase NFR statements. Considerable additional work is needed to improve this grammar.

The following rule fragment shows a small section of the top level grammar in which a recognizable phrase

begins with an optional conjunction, followed by up to three unstructured words, and an optional “The”. Following these optional terms, that represent typical lead-ins to statements, the tool listens for an indicator term.

```
<RULE NAME="NFR" TOPLEVEL="ACTIVE">
  <P>
    <O>
      <RULEREF NAME="Conj" />
    <O>
      <DICTATION MAX="3" />
    </O>
  </O>
  <O>
    <RULEREF NAME="The" />
  </O>
  <RULEREF NAME="Term" />
  <O>
    ...
  </P>
</RULE>
```

Other combinations of indicator terms, standard parts of speech, and unstructured dictation are similarly defined in the grammar. Although this approach can become rather heavy-weight, it was used to support the initial experiments reported in this paper.

4.2 Classification of Speech Detected NFRs

Speech recognition introduces an additional factor into the NFR classification algorithm related to the certainty or uncertainty that a detected word is actually the hypothesized word. Due to natural variations in speech, the same word may be more or less recognizable when spoken by different people, or even from one utterance to the next by the same speaker. For example the word “authorized” might be detected with a 90% certainty in one statement, but with only 60% certainty in another one. The basic detection algorithm shown in equation 2 is therefore extended to incorporate probability values. The following formula shows that in the speech detection algorithm, the probability of a given requirement belonging to a specific quality type is calculated by incorporating the product of term probability and indicator weight for all terms found in the statement.

$$Pr_Q(R) = \frac{\sum_{t \in R \cap I_Q} Pr_Q(t) \times Pr C(t)}{\sum_{t \in I_Q} Pr_Q(t)} \quad (3)$$

Cleland-Huang et al compared the effectiveness of two classification methods [4]. The first threshold-based approach placed a requirement into any category for which it scored over a predefined threshold value, meaning that an individual requirement could potentially

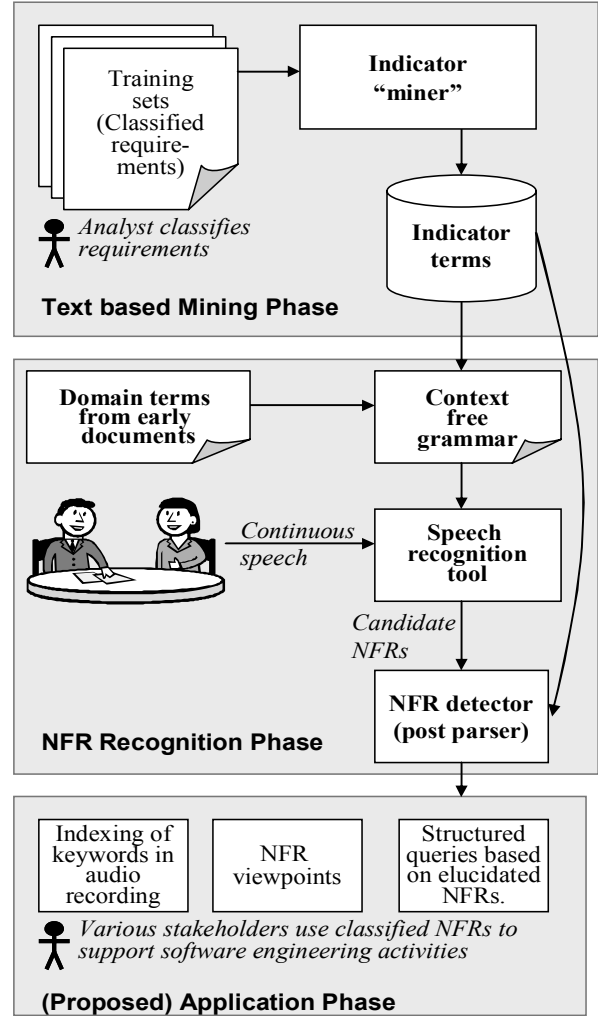


Figure 2. Speech-based NFR recognition

be placed into multiple categories. In contrast, the ‘top’ classification method placed a requirement into the category for which it received the highest score as long as that score was greater than a predetermined value. Prior experimental results [4] indicated that recall was unacceptably low for the “top” classification method, while the threshold approach returned higher recall values at the cost of loss in precision. Therefore the threshold approach was adopted for the speech-based classifier, meaning that a human user would need to filter out incorrect NFR types.

Figure 1a shows an example of the speech recognition system processing an NFR in real-time. Although, the recognition is not completely accurate the system does correctly classify the statement as a **performance** requirement. The statement is however also incorrectly classified as a **usability** requirement, based on the incorrect detection of the word “used” in place of “needs”. In Figure 1b, the sentence “The product manager

Table 2. Stakeholder statements and their classifications

#	Statement	Quality Type
1	When the analyst requests more information about an artifact the response time needs to be fast.	Performance
2	We should be easily able to add new adapters to the system without needing to recompile or even restart the system.	Usability
3	We need a way to help the user differentiate between true and false links. Could you somehow show the likelihood of a link.	Usability
4	We want to be able to trace into MS Word documents too. Not just to more formal case tools.	Non NFR
5	I'm not too concerned about how much time the indexing takes because this will occur offline.	Performance
6	Displaying a class diagram or other such artifact should be completed within three seconds from the time the request is made by the user.	Performance
7	When can we see a prototype of this? We are working on a tight deadline.	Non NFR
8	Updating the indexes should not take more than 60 seconds for each artifact type, unless you are able to fully automate the process so that the project manager doesn't have to wait for the updates to be completed.	Performance
9	We need to make sure that users can only retrieve links from projects for which they are authorized.	Security
10	We may also need to assign different roles and access permissions within a project.	Security
11	Many of our organizations are interested in this, primarily because we want to achieve compliance to CMMI level 3.	Legal/ Policy
12	Can we prevent everyone except project managers from indexing the files?	Security
13	The project manager should set up access rights for people working on the project.	Security
14	We will need to create a matching word list.	Non NFR
15	All data needs to be encrypted before it is sent outside our organization.	Security
16	Can the user provide feedback about a link? This would simplify future queries.	Usability
17	I think that a request for additional data should be serviced in less than 5 seconds. I guess that it might depend on what type of data you are requesting. It would probably take longer to redraw a sequence diagram than to display a requirement. Don't you think?	Performance
18	Because so many different people would use this system, we would need to have a good help system in place.	Usability
19	The system will be used by software engineers. It should have a conservative look and feel.	Look & feel
20	We already have a pretty good trace matrix in place however it does take us a lot of effort to keep it maintained.	Maintainability

should setup access rights for people working on the project” was correctly recognized and detected as a **security** requirement.

Figure 2 summarizes the new speech-based NFR recognition process. Indicator terms initially mined from text based requirements repositories as well as domain terms mined from early project documents are used to construct the context free grammar. The grammar is used to process the continuous speech representing either live speech or an audio file, in order to detect NFR related statements. The final output of the process is a set of candidate NFR statements that can potentially be used to support the requirements process.

5. Experimental Analysis

To evaluate the ability of the speech based detector, twenty typical stakeholder statements were created that represented security, performance, usability, look-and-feel, and functional requirements. These statements are

shown in Table 2. In order to focus on evaluating the speech-based detection algorithm only sentences that were correctly categorized by the text based detector were included.

As this study represented an initial analysis of the technique, we included only three adults for whom American English was their first language. These participants were asked to read the statements into a headset microphone, using intonation similar to that used in ordinary conversation. The statements were recorded in .wav files which were then played back into the tool for NFR recognition. The results are reported in Table 3. The columns labeled P1 represent the results obtained using the grammar described in section 4.1, while the columns labeled P2 reflect the results obtained following a second parse through the detected sentences to retrieve indicator terms that had previously been captured in blocks of free dictation and had therefore not previously contributed to the classification algorithm. In the results table a cell is marked with a “1” to represent the correct classification of a statement. An NFR is considered

Table 3. NFR detection results

Statement	Speaker 1		Speaker 2		Speaker 3	
	P1	P2	P1	P2	P1	P2
1	0	0	1	1	1	1
2	1	1	0	0	0	1
3	0	0	0	0	1	1
4*	0	0	1	1	1	1
5	0	0	1	1	1	1
6	0	1	0	0	1	1
7*	1	1	1	1	0	0
8	0	1	1	1	1	1
9	1	1	0	1	0	0
10	1	1	1	1	1	1
11	0	1	1	1	1	1
12	0	0	0	1	0	0
13	1	1	1	1	1	1
14*	1	1	1	1	1	1
15	0	0	0	0	0	0
16	0	0	1	1	1	1
17	1	1	0	0	1	1
18	1	1	1	1	1	1
19	0	1	0	1	1	1
20	1	1	0	0	1	1
Total correct	9	13	11	14	15	16
Percentage	0.45	0.65	0.55	0.70	0.75	0.8

* Non-NFR statements

correctly classified if the speech detector recognizes it as a candidate NFR of its correct type, while a non-NFR is considered correctly classified if it is rejected from all NFR categories.

The results indicated that following the second parse, the three speakers exhibited accuracy rates of 65%, 55%, and 80% respectively. The results are summarized in the graph shown in Figure 3. Furthermore, the tool was able to correctly reject non-NFR statements 78% of the time. This ability is important, because it suggests that the tool can filter out most non-NFR statements from stakeholders' utterances. Although not perfect, these results indicate the feasibility of creating a grammar capable of recognizing NFRs and rejecting non-NFR related statements.

To compare the results obtained using a context free grammar with those obtained without it; the recordings were reprocessed in dictation mode. The results obtained from Speaker 1 are depicted in Figure 4 and most notably show the inability of the tool to construct meaningful sentences or to capture many of the key indicator terms. The results appear more as random phrases, however in certain cases the free dictation managed to pick up terms

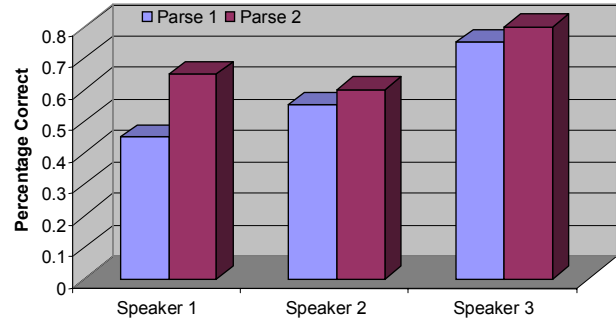


Figure 3. NFR Detection by speaker

that were not picked up when the grammar was used. This is in fact the reason that the second parse of the sentences detected by the grammar improved results.

5.1 Detection vs. understanding

By analyzing the results of the experiment, it became quickly apparent that even though the indicator terms enabled many of the statements to be correctly categorized, in many cases the real meaning of the statements was entirely lost. The general speech detection accuracy deteriorated when statements were spoken in a continuous form. For example the simple statement "When the analyst requests more information about an artifact the response time needs to be fast", was easily detected when spoken as an individual statement in Figure 1a. In contrast when included in continuous speech, this same statement was recognized correctly when spoken by speaker 2. For speaker 1 it was recognized as "where they analyst reports more information about an artifact routes for us and ease the fast" and for speaker 3 as "when the analyst request more information about north actor response time used the fast". Clearly, even though the indicator terms were correctly captured, much of the meaning was lost, especially when spoken by speaker #3.

These results revealed a couple of problems with our approach. First there were several cases in which a key indicator term was found in the free-form dictation part of the text, even though the tool had failed to recognize it as a term. By executing a second parse on the text, it was possible to improve the accuracy of the NFR detection tool. In future work we intend to simplify the grammar and attempt to increase its ability to reconstruct meaningful sentences. By taking the timestamps from the detected sentences sections of the .wav files representing candidate NFR speech-based statements can be isolated. Once continuous speech is deconstructed into individual statements, a series of lighter-weight grammars can be used to parse and categorize the statements. In addition to indicator terms, a parser could also be created to detect domain terms, in order to construct greater meaning from each statement.

6. Conclusions

The accuracy of the NFR classification is limited not only by the completeness and correctness of the indicator terms, but also by the accuracy of the underlying speech recognition. Certain subjects are classified as “goats” [14], in that they are unable to be recognized, or are recognized extremely poorly by the speech recognition system. Furthermore, the type of free form environment that is expected for the NFR detection problem creates numerous additional challenges. However, the possibility of capturing a significant number of NFRs in a semi-automated fashion allows for a much more efficient use of the analyst’s time. One of the advantages of our system is that the phrases associated with a given word can be emitted as a sound file, so that key phrases may be tagged, and keyword searches performed for post-analysis by a human expert.

This paper describes a framework for analyzing conversations in order to extract and classify NFRs using speaker-independent speech recognition. Our approach is based on lightly constraining acceptable utterances by using cross-domain indicator terms, along with specific domain terms mined from previous documentation. In the next phase of our research we will continue to increase the recognition accuracy of the underlying speech recognition system, while using the extracted indicator terms and domain specific information to automatically generate post-discussion questions (either in a written or dialog form). These questions will elucidate the identified NFRs, and provide us a way to query the user in a structured fashion. Part of the research we hope to undertake is to do a side by side comparison between the SAPI system and other ASR systems, such as the open source Phoenix system from CMU and the proprietary Dragon AudioMining system from Nuance [15].

Acknowledgments

This work was partially funded by NSF grants CCR-0306303 and CCR-0447594. Access to project artifacts and additional funding was provided by Siemens Corporate Research and Siemens Logistics and Automation plant.. We also thank DePaul faculty member David Stone for his “sound” advice.

References

- [1] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers 2000.
- [2] S. Robertson and J. Robertson, *Mastering the Requirements Process*, Addison-Wesley, 1999.
- [3] P.Grünbacher, B. W. Boehm, “EasyWinWin: a groupware-supported methodology for requirements negotiation”, ESEC /

a data request for additional data
sure the service of a supporter
data might entail updated annually<
it would only take longer.
and see the state around the
spoiler or anything
because of a different people use
the system will be used percentages
sure that its Irvine law and
we already have a pretty good
ways however it goes into supported
where they analyst reports more information
if she the ease legal entities
would only shaving a sound card
if you wait help these should
Jia late in through false lines
could ease of handling the only
legality the able to trace Indiana's
that is Jamil for more days

Figure 4. Subset of speaker #1’s statements captured using free dictation mode

- SIGSOFT Foundations on Software Eng., 2001 pp.320-321
- [4] J. Cleland-Huang, R. Settimi, X.Zou, and P. Solc, “The Detection and Classification of Non-Functional Requirements with Application to Early Aspects”, IEEE International Conf. on Requirements Engineering, St Paul, MN, September, 2006.
- [5] D. Jurafsky and J. Martin, *Speech and Language Processing*, Prentice-Hall, Englewood Cliffs, NJ, 2000.
- [6] V. Kulyukin and A. Steele, "Input Recognition in Voice Control Interfaces to Autonomous Agents," in *ILC2002*, San Francisco, CA, 2002.
- [7] V. Kulyukin and A. Steele, "Instruction and Action in the Three-Tiered Robot Architecture," in *ISRA'2002*, Toluca, Mexico, IEEE, 2002.
- [8] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman, “Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models”, *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 11, pp. 1870-1878, November 1990
- [9] K. F. Lee., Hon, H. W. and Reddy, "An overview of the SPHINX speech recognition system." *IEEE Transactions on Acoustics, Speech, and Signal Processing*38(1): 35-46 1989.
- [10] Sphinx, “The CMU Sphinx Group Open Source Speech Recognition Engines”, <http://cmusphinx.sourceforge.net/html/cmusphinx.php>.
- [11] SAPI, “Microsoft SAPI 5.1”, <http://microsoft.com/speech>.
- [12] L. R. Rabiner, “A tutorial on Hidden MarkovModels and selected applications in speech Recognition”, *Proc. IEEE*, 77(2), Feb. 1989, pp. 257-285.
- [13] X. Zou, R. Settimi, J. Cleland-Huang, “Phrasing in Dynamic Requirements Trace Retrieval”, International Computer Software and Applications Conference (COMPSAC), September 2006.
- [14] G. Doddington, *et al.* “ Sheep, Goats, Lambs and Wolves: An Analysis of Individual Differences in Speaker Recognition Performance”, ICSLP’98, Sidney, Australia, November 1998.
- [15] Nuance, “Dragon AudioMining”, <http://www.nuance.com/audiomining/>.