UNIVERSITY OF GOTHENBURG

Speech Recognition for Noisy Environments

Feasibility of Voice Command in Construction Settings

Bachelor of Science Thesis in Software Engineering and Management

ARASH AKBARINIA

JAVIER VALDEZ MEDRANO

RASHID ZAMANI

University of Gothenburg
Chalmers University of Technology
Computer Science and Engineering
Göteborg, Sweden 2011

Speech Recognition for Noisy Environments

Feasibility of Voice Command in Construction Settings

ARASH AKBARINIA

JAVIER VALDEZ MEDRANO

RASHID ZAMANI

Examiner: Helena Holmström Olsson

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden 2011

Speech Recognition for Noisy Environments

Feasibility of Voice Command in Construction Settings

Arash Akbarinia
akbarinia.arash@gmail.com

Javier Valdez Medrano
xavier@buzmay.com

Rashid Zamani
rashid.z@gmail.com

IT University of Gothenburg, Software Engineering and Management. Gothenburg, Sweden

CHALMERS | GÖTEBORGS UNIVERSITET

Abstract

* * * * * * * * * * * *

People can comprehend speech even in noisy environments. Yet, the same task for machines still remains to be an elusive ambition. In this paper, by implementing a speech recognition prototype as proof of concept for Volvo Construction Equipment, we illustrate possibility of voice-commanding construction machines in heavy noisy environments. The findings of our research are not limited to Volvo Construction Equipment, and this paper can be studied as a guideline for boosting noise robustness of speech recognition applications.

Categories and Subject Descriptors D.2.4 [*Software Engineering*]: Software/Program Verification: Correctness proofs; J.1 [*Computer Applications*]: Administrative data processing; Manufacturing; J.7 [*Computer Applications*]: Computers in other systems: Command and control;

Keywords: *speech recognition, noise robustness, voice command, machine learning*

1 Introduction

DREAM of machines that can understand human speech has been around from the 13^{th} century and during the last eight decades extensive research was conducted to fulfil this dream. Although great discoveries and advancements are accomplished in this field, the ultimate goal of naturally communicating with machines seems far to fetch [4]. Speech recognition (SR) is a very easy task for human beings and happens subconsciously, but the fact is, the brain processes many factors prior to recognition of speech. Researchers have strived to imitate similar processes to facilitate automatic speech recognition (ASR). However, the task revealed to be extremely hard. Subconscious activities – for instance, considering speech context, checking syntax and semantics, linking acoustic-phonetics, and ignoring background noises – demand complex calculations and still require further research.

Current existing SR applications do not work efficiently in noisy environments. To illustrate this incompetence, prior to the start of our research, we conducted a simple SR experiment on two existing applications – Android 'Speech Input', and Windows 7 'Speech Recognition'. We measured accuracy of mentioned applications under two environments – i.e. quiet and noisy. Both applications performed well in the quiet environment, whereas in the noisy one they showed a considerable amount of inaccuracy. Refer to Appendix B for the details of this experiment.

Robustness to noise and other external artefacts of speaking remains a challenge that is being addressed by interdisciplinary researchers – Signal Processing, Pattern Recognition, Natural Language, and Linguistics [4]. Currently, Volvo Construction Equipment is considering adding SR feature to their construction machines. Thus, by utilising existing techniques, we engineered an SR prototype to investigate our null hypothesis: "recognising speech accurately is not feasible in heavy noisy environments".

Design research is the approach we followed in this study, as it is suggested by Vaishnavi et al. [21] for synthetic disciplines. By reviewing literature, we learnt about current state-of-the-art. By studying existing frameworks, we evaluated current state-of-the-practice. Based on these findings, we implemented the prototype; and in order to verify our null hypothesis we performed various system experiments in different environments. And lastly, by statistically evaluating results of those experiments, we measured the accuracy ratio of our prototype, which helped us to falsify the null hypothesis.

Benesty et al. [4] present different issues that SR is facing, and as it can be observed in figure 1, Becchetti et al. [3] categorise those challenges into four main axes: (i) inverse of the available computational power, (ii) variability of speaker, (iii) complexity of dialogue or size of the vocabulary, and (iv) acoustic speech quality. Recognition is simpler when approaching the origins of the axes. In this research we focus on the "acoustic speech quality", and we strive to show possibility of SR in noisy environments. Our contribution is merely toward noise robustness challenge of SR, since that is the main concern of Volvo Construction Equipment. Therefore, we minimised significance of the other three axes by only: (i) supporting limited number of words – listed in Appendix C, (ii) using powerful laptops, and (iii) primarily focusing on recognising a unique speaker.

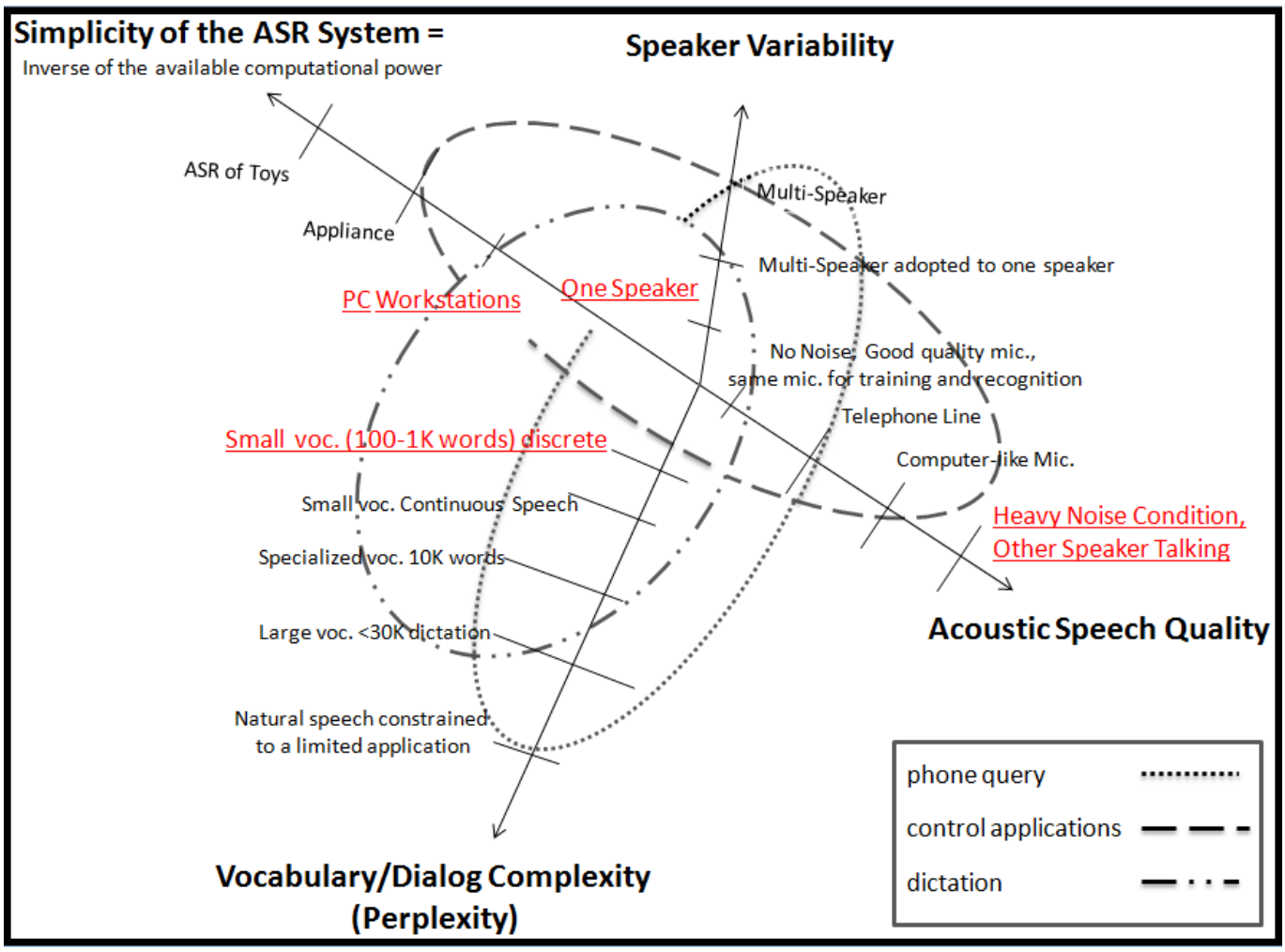


Figure 1: Speech recognition problems and applications [3]. Underlined labels show characteristics of our prototype and contribution of this paper on each category.

In this paper, we present that even in heavy noisy environments, construction machines can potentially be commanded by speech. We speculate four different elements: (i) acoustic model (AM), (ii) speech quality, (iii) language model (LM), and (iv) microphone characteristics. And we present the influence of each element on noise robustness.

2 Research structure

Research can be very generally defined as an activity that contributes to the understanding of a phenomenon [16] [17]. This phenomenon is typically a set of behaviours of some entities that are found interesting by the researcher or an industry [21]. Mapping this onto our bachelor thesis, the phenomenon we are striving to understand is feasibility of SR – in form of voice command (VC) – in heavy noisy background environments. The findings of this research is naturally going to be interesting for industries that are planning to develop similar applications, as well as the research community in the field of SR.

Booth et al. [5] argue that each discipline has standardised research methodologies for collecting and reporting evidence. Following such methodologies guarantees that the research being conducted is reliable. That is why we decided to follow design research, which is a frequently practised technique in computer science and engineering disciplines. This methodology is a recognised approach to understand, explain and improve engineering artefacts, such as software algorithms [21].

In the following two subsections we outline the settings and process of our research.

1 Research setting

We conducted this research as our Software Engineering bachelor thesis at IT university of Gothenburg. The addressed industrial problem was proposed by Volvo Technology. They are interested to learn whether it is feasible to command construction machines – such as wheel loader and excavator – by voice in heavy noisy environments. Subsequently, we implemented a prototype to verify this possibility.

We implemented the prototype in ANSI C programming language on standard computers[1] running a Unix operating system, and therefore system resources such as lack of mem-

[1] 2.1 GHz AMD Processor and 4.00 GB RAM

ory or computation power was not a constraint – the prototype was not an embedded system. The vocabulary size was not an issue, due to the fact that the number of words to be recognised was limited to a few arbitrary chosen commands from the project acquirer. These commands facilitate controlling the body and bucket of construction machines. Refer to Appendix C for the complete list. Finally, Volvo Technology and we agreed to lower priority of the speaker variability factor, since our main focus was on speech quality. Therefore, the prototype ought to primarily work with a unique speaker – male non-native English speaker.

2 Research process

Vaishnavi et al. [21] recommend design research for disciplines that are synthetic. Hence, we chose the design research methodology, because our prototype is in-line with "Product Design" and close to synthetic research category. We designed a product, which included construction and evaluation of an artefact that according to Vaishnavi et al. [21] led us to build knowledge. In order to do that, we performed the following five steps – (i) systematic literature review, (ii) existing frameworks evaluation, (iii) prototype development, (iv) system experiment, and (v) evaluation – which are also illustrated in figure 2:

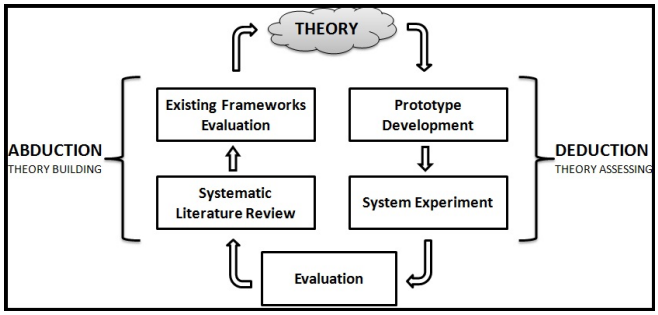


Figure 2: Reasoning in design cycle [21].

In the five following subsections, we describe each of the five steps we followed in our research process. It must be noted that we performed all steps iteratively. As Vaishnavi et al. [21] suggest knowledge is generated and accumulated through action. In our case, studying, implementing, testing and judging the results helped us to improve the prototype.

i Systematic literature review

We systematically reviewed literature to discover current state-of-the-art and state-of-the-practice in different SR components – e.g. different techniques to train AM or reducing noise level. Booth et al. [5] categorise literature sources into three different types. The below list outlines our sources mapped onto this classification:

- Primary – raw materials of our research topic, i.e. algorithms and techniques, belong to this category.
- Secondary – researchers' related works, i.e. articles, books, and journals, belong to this category.
- Tertiary – reference works in our subject, i.e. encyclopaedias, belong to this category.

SR is a fairly young technology, and it is evolving constantly. According to Booth et al. [5], journals and articles are concrete sources of information for these types of technologies that are changing rapidly. Many articles and journals exist in this field; it is counter-productive to read all of them in detail. Therefore, as Brusaw et al. [6] and Galvan [10] suggest, by skimming through abstracts, introductions, and conclusions; we realised which articles require in depth studying.

Based on our secondary sources findings, we reached a better understanding about algorithms and libraries that suited our requirements the most. During the second and third steps of our research process – evaluating existing frameworks and prototype development – we gained knowledge from our primary sources. Whenever required during all steps of our research, we referred to tertiary sources to expand our vision.

ii Existing frameworks evaluation

There are already many free and proprietary SR frameworks and libraries. In this step of research, by reading forums and studying library documentations, we investigated which one of the free libraries is more suitable to use in our prototype and build the prototype on top of that. As explained earlier, our focus was on speech quality; therefore noise robustness feature was our main interest in evaluation of libraries. Based on the lessons learnt from the systematic literature review, we realised which type of noise reduction algorithms – filtering techniques, spectral restoration, and model-based methods [4] – is more suitable for our environment. Consequently, we looked for that algorithm in existing libraries and chose the one, which suited our requirements.

iii Prototype development

With the knowledge gained from steps one and two of our research process, we started implementing a prototype by following a test driven development (TDD) approach. It must be pointed out that the purpose of this research was not to develop a new SR algorithm, but rather to combine existing solutions to satisfy project requirements. As it was explained before, our focus in development was on speech quality and not on the other three SR challenges.

iv System experiment

In this step, we tested the implemented prototype in different environments with variety of background noises. Each command was pronounced by a unique speaker in order to check the accuracy of recognition. Prior to start of the first iteration, a few samples of all commands were recorded in Volvo Construction Equipment working environment with the same microphone that we used for prototype development.

Basili et al. [1] state that a good experiment is replicable. Therefore, we recorded all the experiments in order to re-test them with future versions of our prototype. This is in-line with the fact that any scientific theory must be: (i) falsifiable, (ii) logically consistent, (iii) at least as predictive as other competing theories, and (iv) its predictions have been confirmed by observations during tests for falsification.

If we map our research onto validity suggested by Campbell and Stanely [7], our factor of interest is speech quality.

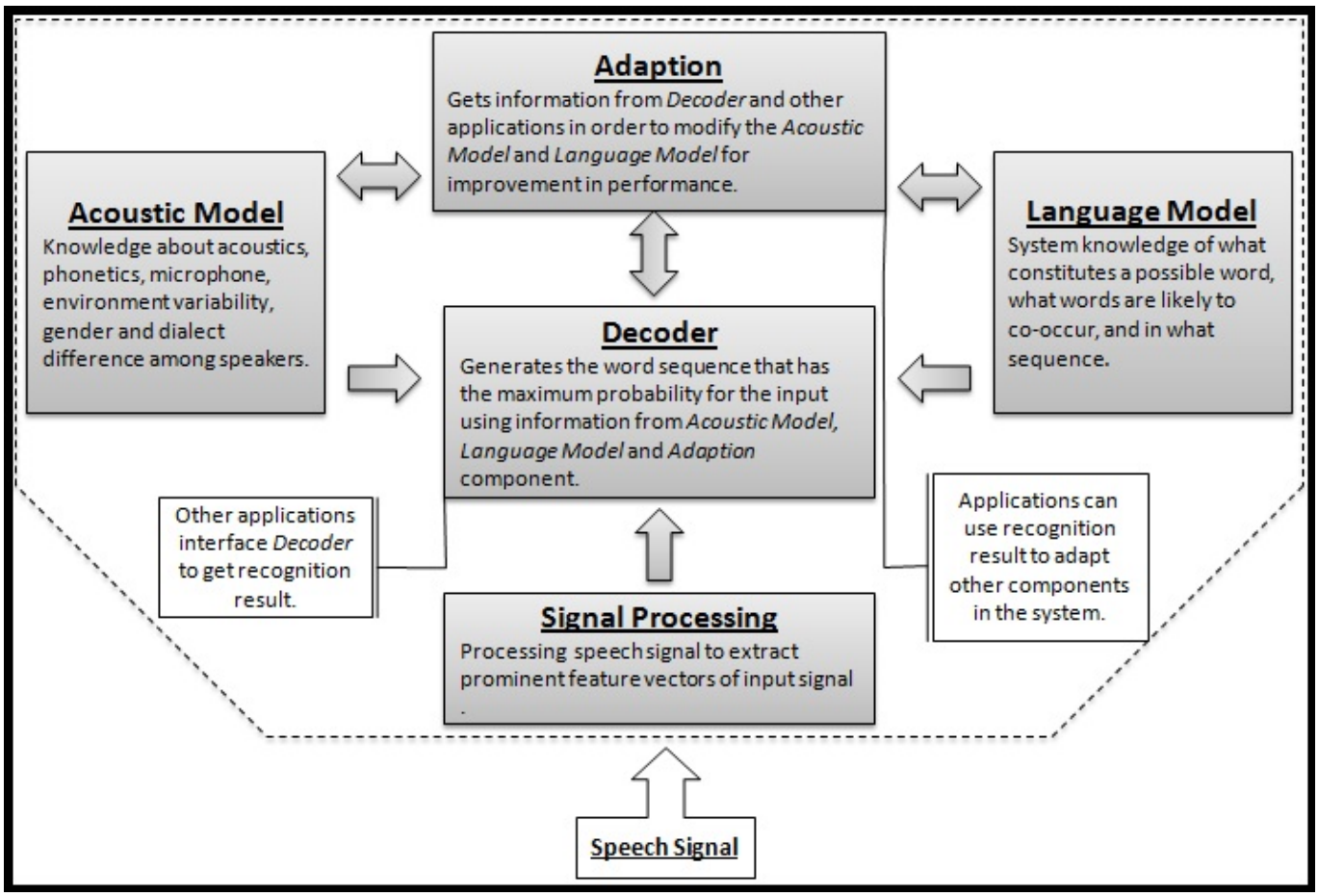


Figure 3: Basic system architecture of a speech recognition system [12].

Therefore, our internal validity was to check whether the implemented prototype worked properly in different construction settings of Volvo Construction Equipment. Our external validation was to check accuracy ratio of VC in similar heavy noisy environments. Finally, we presented our results statistically to prove conclusion validity suggested by Cook and Campbell [8].

As suggested by Basili et al. [2], experiment has a learning process. That is why we decided to perform our study iteratively, in order to modify all steps based on the findings of each iteration. For instance, at the beginning of our experiment, we did not know the exact criterion for experiment interpretation. However, after the first iteration, the experience lead us to build a more explicit vision. We also modified our means of data collection and analysis based on the lessons learnt from each iteration, to ensure the collected data are comparable across different projects and environments [2].

As suggested by Creswell [9], we tried to control independent variables – speaker, commands, microphone, computer, background noise environments – and check the treatment – our prototype. The dependant variable was our prototype accuracy, which we measured in our experiments.

v Evaluation

For each experiment configuration, we statistically calculated the number of commands recognised correctly to measure the accuracy ratio of that configuration. Subsequently, we compared the extracted accuracy ratios to conclude which configuration meets Volvo Construction Equipment requirements. Following to that, we argued whether the null hypothesis was verified or falsified.

3 Background

SR can be employed in many different types of applications, such as: (i) rich transcription, which is not only SR, but also speaker identification; (ii) voice command (VC), in which isolated words are recognised; (iii) audio search, i.e. searching for quotes in audio files; and (iv) structuring audiovisual databases, for example detecting whether a sound is from a formal meeting, a news broadcast, or a concert. Our prototype can be categorised as VC, which has its own difficulties. For instance, because VC applications are usually embedded systems – e.g. commanding your navigation system and mobile-phone – computational power can be a constraint. Additionally, VC applications are sometimes very critical; therefore, robustness is very important. Consider commanding aeroplanes or cars; one mistake can endanger peoples' life.

As it is demonstrated in figure 3, according to Huang et al. [12], a typical SR system consists of five components: (i) *Signal Processing*, (ii) *Decoder*, (iii) *Adoption*, (iv) *Acoustic Model*, and (v) *Language Model*. In this study, we concentrate on four different elements: (i) acoustic model and par-

ticularly its training, which in figure 3 can be mapped onto the connection between *Adaptation* and *Acoustic Model*. (ii) Speech quality, which is employed before speech signal is passed to *Signal Processing*. (iii) Language model, which naturally belongs to *Language Model* component. And (iv) microphone characteristics, which influences quality of *Speech Signal*.

In the following two subsections, we first describe the problem that noise causes in SR. Second, we explain four potential solutions for noise robustness.

1 Problem with noise

Recognition of speech in construction settings is very difficult, due to the loud noise produced by heavy machines and wind. Huang et al. [12] categorise main sources of distortion into (i) additive noise, and (ii) channel distortion. The former is caused by background noise, such as engine of a lorry, other speakers' voices, or wind sound; and the later can be caused by reverberation, the frequency response of a microphone, or the presence of an electrical filter in the A/D circuitry.

In the following three subsections, we first characterise additive noise and channel distortion. Next, we describe how both types of noise contaminate speech signal.

i Additive noise

This type of noise is divided into stationary and non-stationary. Stationary noise has a power spectral density that does not change over time, for instance the noise produced by a computer fan or lorry engine. Non-stationary noise, caused by i.e. door slams, radio, television, and other speakers' voices, has statistical properties that change over time [12].

ii Channel distortion

If both the microphone and the speaker are in an anechoic chamber or in free space, a microphone picks up only the direct acoustic path. However in practice, in addition to the direct acoustic path, there are reflections of walls and other objects in the room [12].

iii Speech contamination

Both types of distortion contaminate the speech signal and change the data vectors representing speech; this will cause a mismatch between the phoneme of training and operating environments. Therefore, as it was explained in the introduction section, speaking environment is one of the most important factors in accuracy ratio of ASR. In this research, we look into additive noise and how to overcome the challenges they impose.

Huang et al. [12] present that the error rate of machines, unlike humans, increases dramatically when the environment becomes noisy – in 10-db Signal-to-Noise Ratio (SNR) accuracy ratio was dropped two times than clean speech. In a study by Gunawardana et al. [18], word accuracy for the Aurora 2.0 was rapidly degraded even at a mild 20-db SNR – the system produced more than fourteen times as many errors compared to clean data. Considering the fact that 10-db is light whisper and 20-db is condition of a quiet living room; one can imagine difficulty of SR in noisier environments, such as busy city streets – 70-db – or power tools – 110-db. Increasing SR robustness in noise is a challenge that according to Benesty et al. [4] is currently being addressed in the fifth generation of SR research.

2 Solution

According to Huang et al. [12], one of the best solutions for noise robustness is to train the AM with data gathered from the operating environment. In this method, the Hidden Markov Model (HMM)[2] of AM is trained for that acoustic environment, and the noisy speech is decoded without further processing. This is known as *matched condition* training.

Another solution that Huang et al. [12] suggest is to clean noisy features, which can be combined by training of HMM. It has been demonstrated that feature normalisation alone can provide many of the benefits of noise robustness specific algorithm. Because these techniques are easy to implement and provide impressive results, they should be included in every noise-robust SR system.

On top of those two solutions, constructing an adapted language model that contains required dictionary and grammar is considered to be very beneficial for noise robustness. Finally, noise cancelling microphones can be utilised to reduce noisy features from speech signal.

In the following subsections, we categorise existing solutions into four categories: (i) acoustic model, (ii) speech quality, (iii) language model, and (iv) microphone characteristics. And in each subsection, we shortly present the previous works on that area.

i Acoustic model

ASR is fundamentally a pattern-matching problem. The best way to train any pattern recognition system is to train it with samples that are similar to those it has to recognise later. According to Huang et al. [12], by acoustic model training[3] (AMT), application can modify parameters to better match variations in microphone, environment noise, and speaker. One of the AMT techniques is the forward-backward Baum-Welch algorithm, which according to Expectation-Maximisation (EM), it guarantees a monotonic likelihood improvement on each iteration, and eventually the likelihood converges to a local maximum.

Taken to extreme, the AMT can go to the lowest level of language structure, and single-utterance retraining can be performed. The first step is to extract exemplar noise signals from the current noisy utterance. This is then used to artificially corrupt a clean training corpus. Finally, an utterance specific AM is trained on this corrupted data [4].

'Explicit noise modelling' is a recommended algorithm to adapt HMM to non-stationary noise. Dedicating whole-word

[2] Explaining HMM is not within the scope of this paper. Refer to Rabiner et al. [19] for further studying.

[3] In some literature it is also known as "acoustic model adoption".

garbage models can bring some of the advantages of an HMM noise model without the additional cost of a three-dimensional Viterbi search [12]. Ward et al. [22] show the significant improvement of HMM utilising 'noise words' comparing to the one without. In this technique new words are created in the AM and LM to cover non-stationary noises such as lip smacks, coughs, and filler words such as 'uhm' and 'uh'. These nuisance words can be successfully recognised and ignored during non-speech regions, where they tend to cause the most damage. Figure 4 illustrates different steps of 'explicit noise modelling'.

Step 1: Augmenting the vocabulary with noise words (such as ++SMACK++), each composed of a single noise phoneme (such as +SMACK+), which are thus modelled with a single HMM. These noise words have to be labelled in the transcriptions so that they can be trained.
Step 2: Training noise models, as well as the other models, using the standard HMM training procedure.
Step 3: Updating the transcription. To do that, convert the transcription into a network, where the noise words can be optionally inserted between each word in the original transcription. A forced alignment segmentation is then conducted with the current HMM optional noise words inserted. The segmentation with the highest likelihood is selected, thus yielding an optimal transcription.
Step 4: If converged, stop; otherwise go to Step 2.

Figure 4: Noise Modelling Algorithm [12]

ii Speech quality

Speech enhancement techniques rely on differences between characteristics of speech and noise. Thus, the first step when confronted with a particular noise problem is to identify the noise characteristics [11]. Following to that, based on the noise characteristic, either a proper filter must be selected or a new filter must be designed to clean input signals. In signal processing, filters are devices or processes intended to clean the signal from unnecessary features. Although, based on different characteristics – i.e. analogue or digital, linear or non-linear, discrete-time or continuous-time, and passive or active – filters are categorised into different classifications. In reality, some of these classifications overlap time to time. This is the main reason why there is no simple classification for filters [23].

Linear filtering of digital signals is an essential technique to either improve signal components of interest or to reduce noise components [11]. Elliptic filter is a linear filter famous for noise cancellation. Elliptic filter is designed with band-pass and band-stop behaviour. Band-pass filter allows a certain band of frequency to pass through the filter, while it attenuates the rest. Noises that are outside the frequency range of human voice can be filtered easily by using band-pass filter that only passes the human voice. On the other hand, band-stop filter such as Notch, allows most of the frequency to pass, while it lowers the decibel level of certain frequency range.

In many environments, the noise that SR applications are dealing with is additive [11]. As it was described in the 'problem with noise' section, there are two different types of additive noises: stationary and non-stationary. Stationary noises are almost constant in frequency. Therefore, noise frequency can be estimated during pauses in speech. Additionally, because most of noise energy carries out by one or two dominant frequency region [15]; removing these dominant frequencies, by using Elliptic Notch filter, results in a considerable improvement in SR accuracy ratio [11] [15].

In contrast to stationary noise, characteristics of non-stationary noise vary in time. This implies the use of adaptive system capable of identifying and tracking the noise characteristic [15]. Adaptive filter is a pattern recognition filter, which can be self-adjusted to the noise characteristics of environment. Adaptive filters have had a successful commercial achievement, for instance, high-speed modems, or long distance telephone and satellite communication are equipped with adaptive echo cancellers, allowing simultaneous two-way connection [24]. The generic adaptive filter can be applied in different architectures. The functionality of these architectures can be listed as follow [11]:

- System identification: the adaptive filter is placed parallel to a system, and both systems receive the same input signal. The input-output behaviour of system and filter is identical.
- Inverse system identification: the adaptive filter and a system are placed in series, and use a broadband input signal. This architecture is used for echo and delay cancellation.
- Prediction: adaptive filter predicts the current sample value from past signal values. Filter will replicate the predictable signal components as its output, whereas it will only retain the random, uncorrelated part of the signal.
- Noise cancellation: in this case the desired signal is formed by a signal of interest corrupted by noise. A reference signal of the noise is appropriately modified by the adaptive filter to match the noise once filtered. That reference could be taken from the noise source. After adoption, the output signal will ideally contain only the signal of interest.

iii Language model

According to Huang et al. [12], training LM is equally important as AMT in recognising speech. Including variant pronunciations in LM dictionary, according to speaker's dialect, can improve recognition ratio. For instance, if default pronunciation for 'one' is 'W AH N', but speaker pronounces it as 'HH W AH N'. By appending the second alternative in the LM dictionary, decoder can recognise speaker's pronunciation.

Context-Free-Grammar (CFG) is widely used to specify the permissible word sequences in natural language processing when training corpora are unavailable. It is suitable for dealing with structured command and control applications in which the vocabulary is small and the semantics of the task is well defined [12].

iv Microphone characteristics

Speech quality is influenced by the technology being used in the microphone and its relative position to the mouth [4]. Part

of noise cancelling is usually performed in the microphone itself. Thus, selecting a microphone which is capable of eliminating background noises, can improve noise robustness. Huang et al. [12] suggest that a headset microphone is often needed for noisy environments, although microphone array or blind separation techniques have the potential to close the gap in the future.

According to Sadaoki [20], the principal cause of SR errors is a mismatch between the input speech and AM or LM. The input speech from microphone might not match AM or LM, due to (i) distortion, (ii) electrical noise, and (iii) directional characteristics. Equipping SR application with a microphone that can minimise influence of those three mentioned obstacles, declines the probability of mismatch between input speech and AM or LM.

4 Prototype experiment

In the background section, we described four elements – namely (i) acoustic model, (ii) speech quality, (iii) language model, and (iv) microphone characteristics – that influence noise robustness. In order to assess influence of each element, we finalised the prototype on top of those four pillars and conducted rounds of experiment and evaluation, to investigate the proposed research hypothesis: "recognising speech accurately is not feasible in heavy noisy environments".

In the following three subsections, we first describe the experiment preparations. Second, we outline settings of our experiments, and how we performed the experiment. And finally, in the third subsection, we present results of our experiment.

1 Preparation

We implemented a stand-alone SR prototype in ANSI C programming language, which works under Unix operating systems. Prior to start of the implementation, we studied existing frameworks to find the one that suits our prototype requirements. The results all pointed out to CMU-Sphinx[4], a leading speech recognition tool-kit with various supports for different platforms, developed at Carnegie Mellon University. We checked different AMs included in the framework, and selected the American English HUB4 AM, since it produced higher accuracy comparing to the other models.

PocketSphinx – the C implementation of Sphinx tool-kit, which is suitable for embedded systems – was selected as our speech engine recogniser. We chose PocketSphinx over the Java implementation of Sphinx tool-kit – Sphinx 4 – to smooth the process of transferring the prototype into industrial application, as it was requested by Volvo Technology. Furthermore, since the tool-kit is free software – released as open source with a BSD-style license – it is possible in the future to modify low-level configurations to adjust the application for industrial needs.

[4] http://cmusphinx.sourceforge.net/

In order to train the AM, we used SphinxTrain; explicitly the Baum-Welch algorithm. Although the skeleton of our prototype was structured around the Sphinx tool-kit, we employed different frameworks for noise filtering and reduction, namely (i) The Synthesis Tool-Kit in C++ (STK)[5], and (ii) Audacity[6]. To construct grammar and LM we selected CMUclmtk tool-kit. Finally, SphinxBase handled our audio port communication.

2 Settings

In the first stage of our experiment, we recorded eight different samples, which we used for both AMT and experimentation. All recordings were single-channel – monaural, little-endian, unheadered 16-bit signed PCM audio file sampled at 16000 Hz. And all the collected audio samples had a unique speaker. Four different environments were used for background noise, and in each environment we recorded 34 commands with two different microphones – Peltor[7] and Plantronics[8]. The noise level was approximately 80-db at the most.

After we collected sample audio files, we trained the primary AM in two different branches. One was including the 'explicit noise modelling' and the other excluding that. The order of training was as it is illustrated in table 1.

AM	Microphone	Environment of recording
01	Plantronics	Motorcycle
02	Peltor	Motorcycle
03	Peltor	Vacuum-cleaner
04	Plantronics	Vacuum-cleaner
05	Plantronics	Construction settings I
06	Plantronics	Construction settings II
07	Peltor	Construction settings I
08	Peltor	Construction settings II

Table 1: Recorded samples

We examined each of the recorded audio samples in all the sixteen produced AMs, as well as in the primary one without any training – AM00. In other words, all the samples were inspected in four different configurations. Table 2 illustrates the experiment configuration by a two dimensional matrix.

		Explicit Noise Modelling	
		Including	Excluding
Grammar	Activated	Figure 5	Figure 6
	Inactivated	Figure 7	Figure 8

Table 2: Matrix of experiments.

There are ten columns in each mentioned figure in table 2. The most left column is the name of the recorded sample. 'PL' indicates Plantronics microphone, and 'PE' stands for Peltor one. The string after that shows the recording environment. All other columns indicate one AM, starting from 00 to 08. AM00 is the untrained AM, whereas the rest are trained.

[5] https://ccrma.stanford.edu/software/stk/index.html/
[6] http://audacity.sourceforge.net/
[7] http://peltorcomms.3m.com/
[8] http://www.plantronics.com/

The cells show the percentage of correct recognised commands. The bold borders demonstrate AMs that have been trained in the same environment as the mentioned sample in the row. The dotted pattern cells show the AM was trained with the very same sample as stated in the row. The highest percentage in each row is underlined and the lowest is italic.

After AMT, we chose the best configuration from table 2. Subsequently, we conducted another rounds of experiment after filtering all the recorded samples with different filters. The results of those experiments are displayed in figures 9, 10, 11, and 12.

At the end, the results were analysed from four different points of view to show significance of: (i) acoustic model, (ii) speech quality, (iii) language model, and (iv) microphone characteristics, in noise robust SR applications. Based on the lessons learnt from the analyses, we conducted our ultimate experiment in demo environments of Volvo Construction Equipment.

3 Results

In this section we present the analysed results in four different categories, corresponding to the proposed solutions in the background section.

i Acoustic model

The following steps describe the AMT process[9]:

1. We carefully listened to the recorded audio files, and modified the transcription. For instance, if the pronounced sentence was “BUCKET UP”, and there was a noise between two words, we changed the transcription to “BUCKET ++NOISE++ UP”.

   Step one was performed only when ‘explicit noise modelling’ was included. For the branch that ‘explicit noise modelling’ was excluded, transcription of commands was unchanged. For instance, if the pronounced sentence was “BUCKET UP”, and even if there was a noise between two words, we kept the transcription as “BUCKET UP”.

2. We generated acoustic feature files by using *sphinx_fe*.
3. We converted the *sendump* and *mdef* files, by running *pocketsphinx_mdef_convert*.
4. We updated AM files with *map_adapt*.

Once the AMT was finalised, two branches of eight AMs were created. Following to that, we compared the accuracy ratio of all commands with the two mentioned AM branches. One that was trained including ‘explicit noise modelling’ algorithm, and the other excluding that. The primary results showed an insignificant difference between these two. Therefore, we cannot conclude including ‘explicit noise modelling’ in AMT improves the accuracy ratio.

As it can be observed in figures 5 and 6, the AMs that ‘explicit noise modelling’ was included in their trainings, produced slightly lower accuracy. The average of correct recognised commands in figures 5 and 6 were 48 and 49 per cent respectively. The difference was atomic; therefore no conclusion can be made.

Both of the training branches recognised far more correct commands than the preliminary AM without any training – AM00. Hence, we can conclude AMT can significantly improve noise robustness. This fact can be perceived by comparing the best result of each row – which is underlined – with the first column of each table. For instance as it can be observed in figure 5, in ‘PE - Construction II’ sample 56 per cent of commands were correctly recognised by using AM08, which was trained under the same environment. Whereas, the AM00 recognised only 3 per cent correctly. This fact is applicable to all other samples.

Although, AMT can significantly improve noise robustness, it must be performed carefully. Otherwise, the accuracy of that AM might decline. Our experiment results indicate that the highest accuracy ratio is yield, when the AMT is conducted with samples from the same environment, in which the application is going to be deployed at.

To illustrate this fact, observe both recorded samples with vacuum-cleaner noise, which performed better in AM03 and AM04. Those AMs were trained with the same background noise. This fact is applicable for the majority of other samples, except the ones recorded in motorcycle environment. In which, the results for the AMs trained in that environment produced a similar result to the highest value. For instance, in figure 5, in the first row for ‘PL - Motorcycle’ the highest value is 79 per cent in AM04, while the result in AM01 is 76 per cent, which is essentially identical to the highest.

Therefore, we can still conclude when the application is going to be deployed in a construction setting, the material for AM training must be recorded from the very same environment. Observe the bold bordered cells, which indicate those AMs where trained with the samples from the same environment.

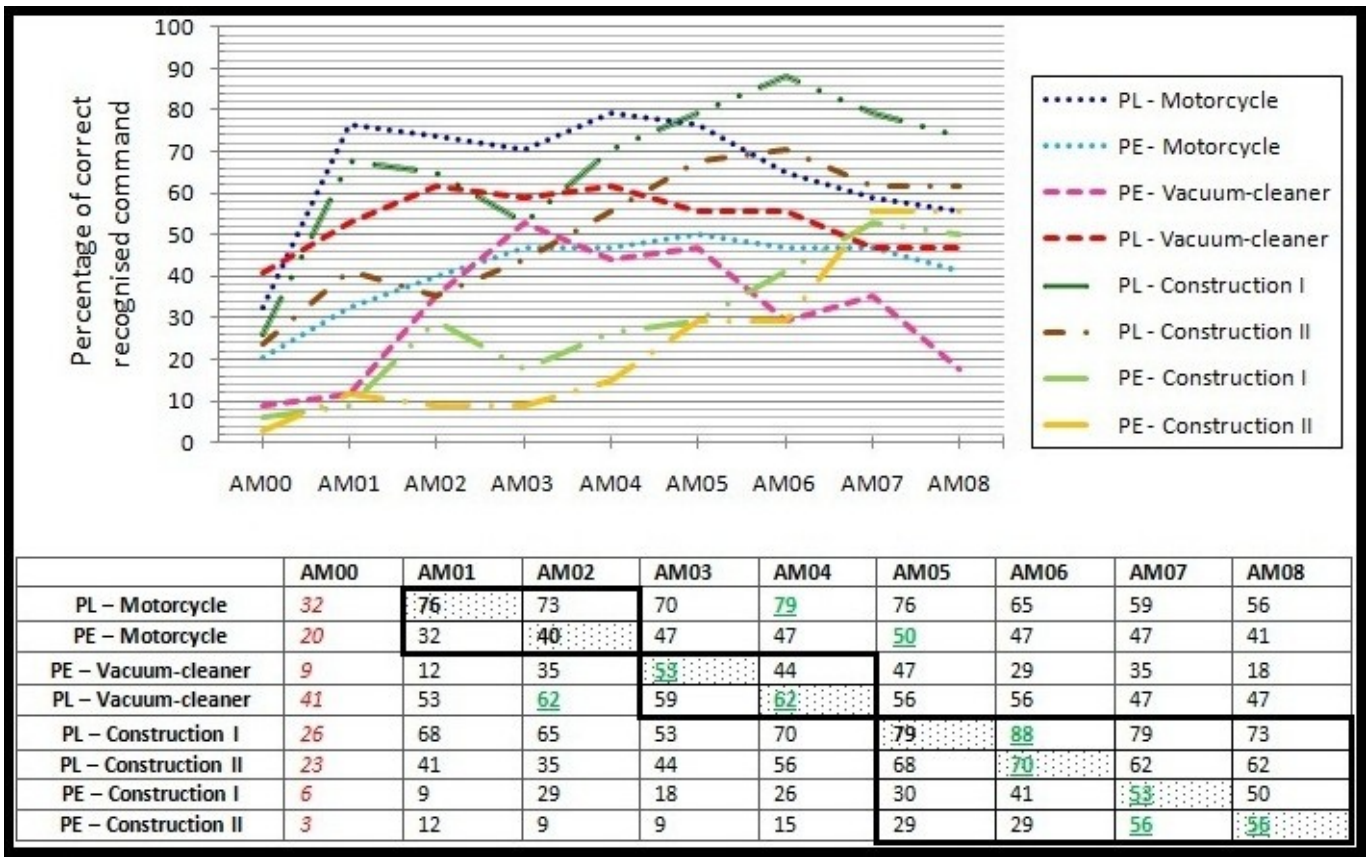
The gathered data also indicates it is better to train the AM with the same microphone that is going to be used for the real application. Even though the microphone factor is not as influential as the environment in AMT, but it still can improve the general accuracy. As an example, ‘PL - Construction I’ that was recorded with a Plantronics microphone, scored better in AMs which were trained with the same microphone – AM05 and AM06 – rather than those that were trained under the same environment but with another microphone – AM07 and AM08. This fact is almost valid for all the other samples. Observe the dotted pattern cells in figures 5, 6, 7, and 8.

ii Speech quality

To examine noise filtering, we selected four different samples that were recorded in two different environments – ‘Vacuum-cleaner’ and ‘Construction II’ representing stationary and non-stationary noise respectively. We applied two different methods – i.e. Notch filter and Audacity pattern recognition ‘noise removal’ feature – to remove noisy features.

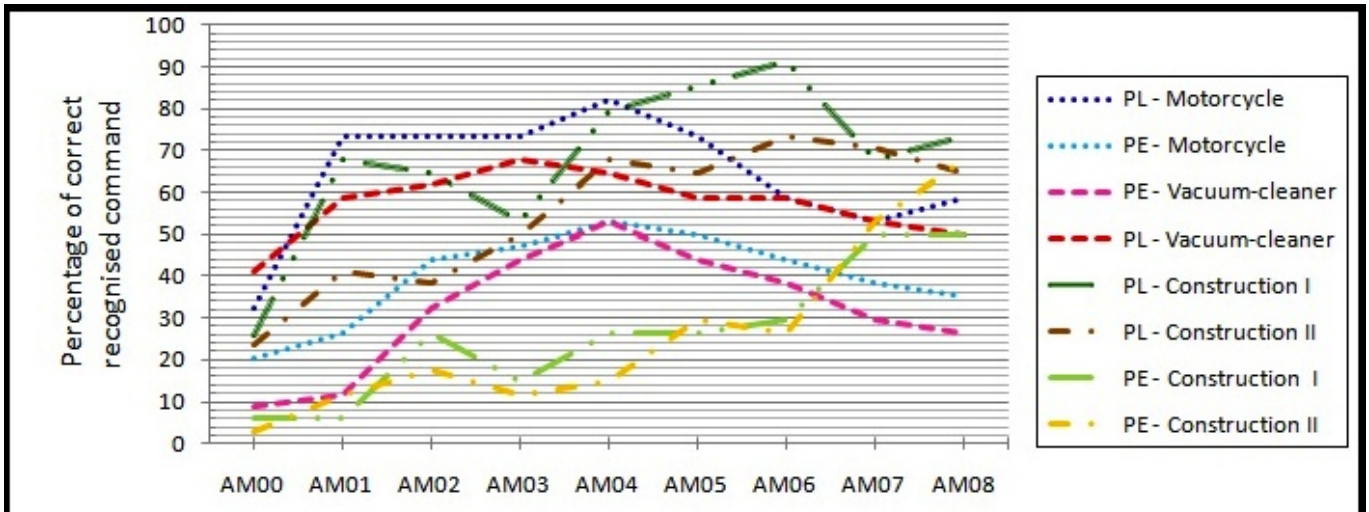
Figures 9, 10, 11, and 12, show the influence of these filters on the accuracy ratio in percentage. Each figure demonstrates whether the employed filter increased or decreased the accuracy ratio of recognised commands. For example,

[9] http://cmusphinx.sourceforge.net/wiki/tutorialadapt



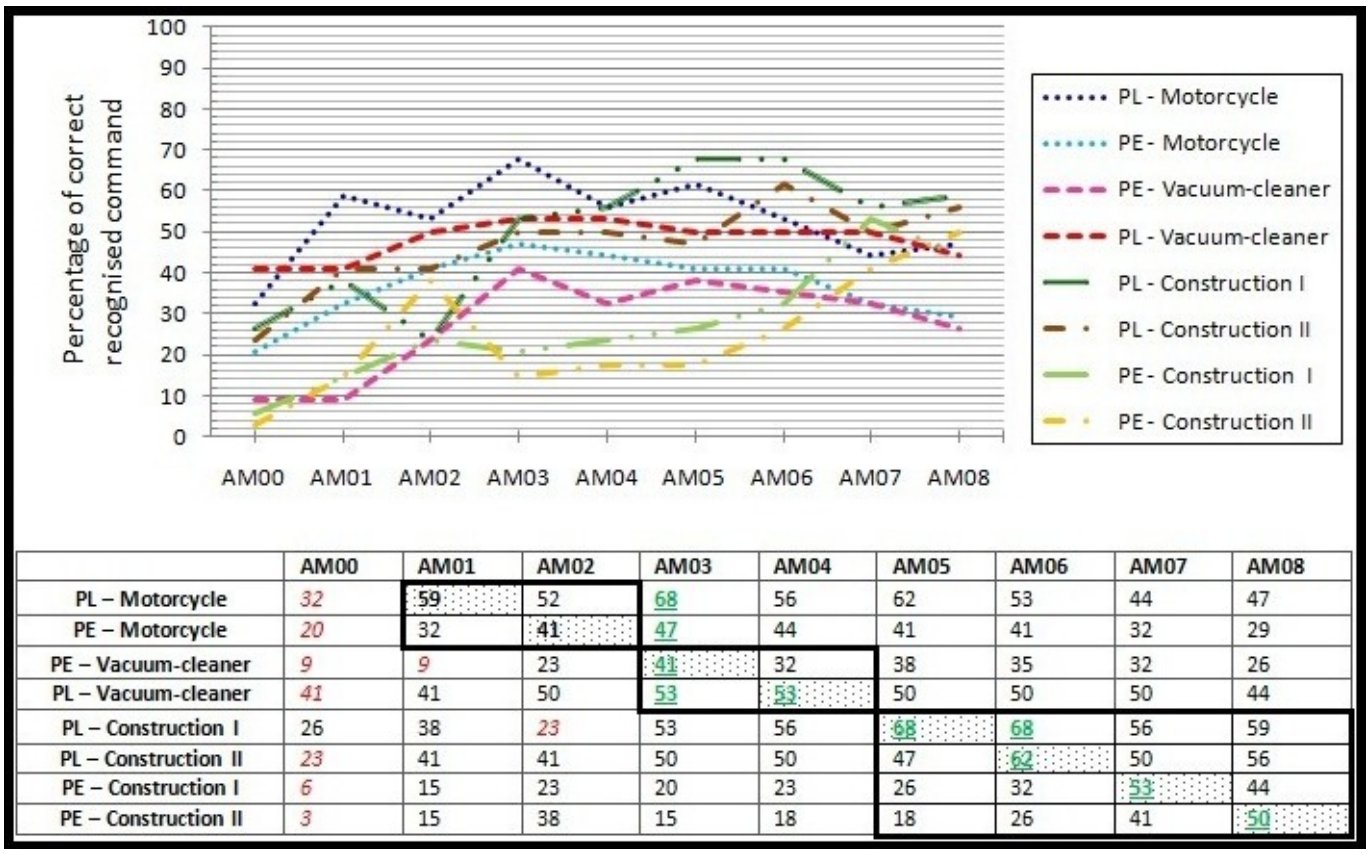
	AM00	AM01	AM02	AM03	AM04	AM05	AM06	AM07	AM08
PL – Motorcycle	32	76	73	70	79	76	65	59	56
PE – Motorcycle	20	32	40	47	47	50	47	47	41
PE – Vacuum-cleaner	9	12	35	53	44	47	29	35	18
PL – Vacuum-cleaner	41	53	62	59	62	56	56	47	47
PL – Construction I	26	68	65	53	70	79	88	79	73
PL – Construction II	23	41	35	44	56	68	70	62	62
PE – Construction I	6	9	29	18	26	30	41	53	50
PE – Construction II	3	12	9	9	15	29	29	56	56

Figure 5: Experiment results for acoustic models trained **including** 'explicit noise modelling'. Language model grammar is **inactivated** in this configuration.



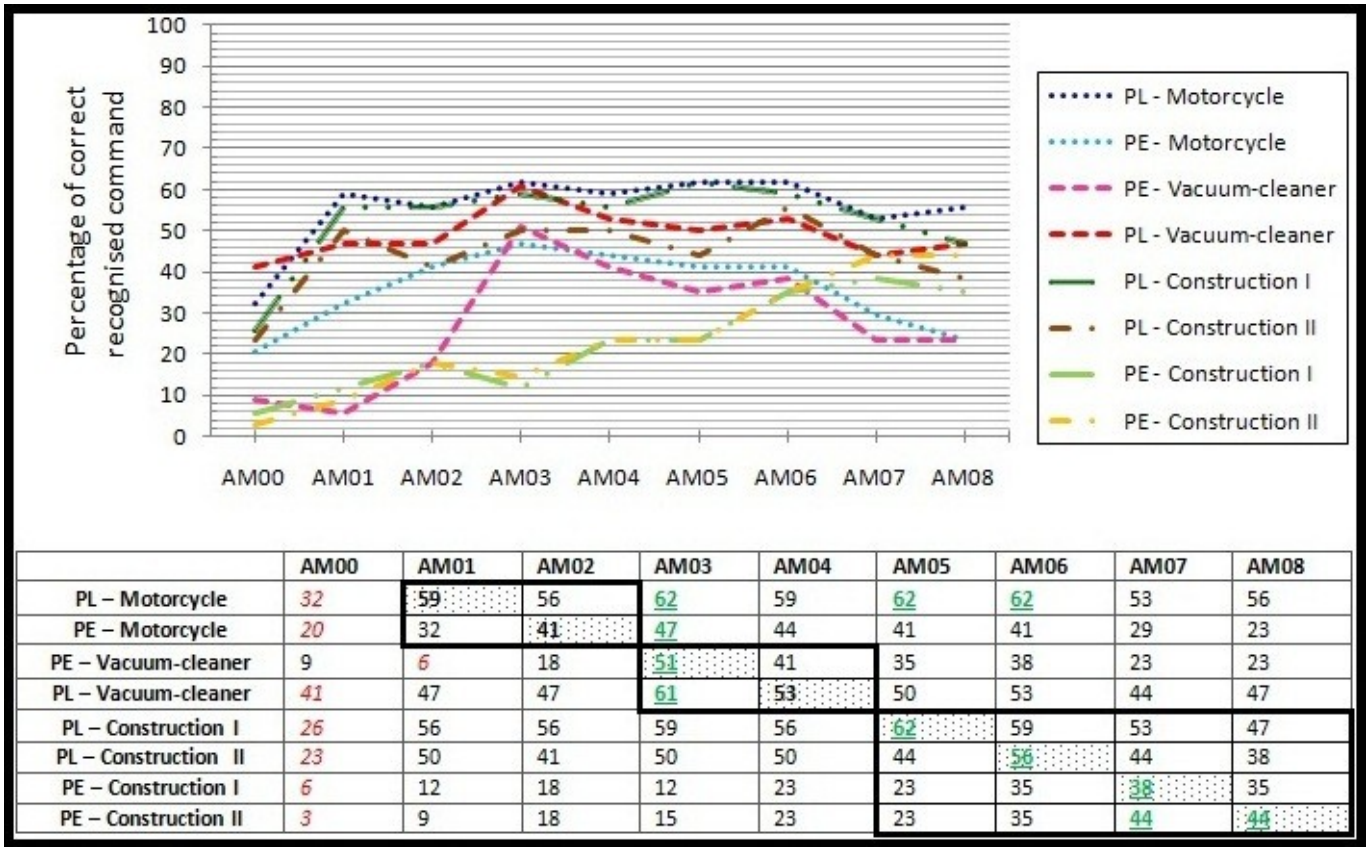
	AM00	AM01	AM02	AM03	AM04	AM05	AM06	AM07	AM08
PL – Motorcycle	32	73	73	73	82	73	59	53	59
PE – Motorcycle	20	26	44	47	53	50	44	38	35
PE – Vacuum-cleaner	9	12	32	44	53	44	38	29	26
PL – Vacuum-cleaner	41	59	62	68	65	59	59	53	50
PL – Construction I	26	68	65	53	79	85	91	68	73
PL – Construction II	23	41	38	50	68	65	73	70	65
PE – Construction I	6	6	26	15	26	26	29	50	50
PE – Construction II	3	12	18	12	15	29	26	53	68

Figure 6: Experiment results for acoustic models trained **excluding** 'explicit noise modelling'. Language model grammar is **inactivated** in this configuration.



	AM00	AM01	AM02	AM03	AM04	AM05	AM06	AM07	AM08
PL – Motorcycle	32	59	52	68	56	62	53	44	47
PE – Motorcycle	20	32	41	47	44	41	41	32	29
PE – Vacuum-cleaner	9	9	23	41	32	38	35	32	26
PL – Vacuum-cleaner	41	41	50	53	53	50	50	50	44
PL – Construction I	26	38	23	53	56	68	68	56	59
PL – Construction II	23	41	41	50	50	47	62	50	56
PE – Construction I	6	15	23	20	23	26	32	53	44
PE – Construction II	3	15	38	15	18	18	26	41	50

Figure 7: Experiment results for acoustic models trained **including** 'explicit noise modelling'. Language model grammar is **activated** in this configuration.



	AM00	AM01	AM02	AM03	AM04	AM05	AM06	AM07	AM08
PL – Motorcycle	32	59	56	62	59	62	62	53	56
PE – Motorcycle	20	32	41	47	44	41	41	29	23
PE – Vacuum-cleaner	9	6	18	51	41	35	38	23	23
PL – Vacuum-cleaner	41	47	47	61	53	50	53	44	47
PL – Construction I	26	56	56	59	56	62	59	53	47
PL – Construction II	23	50	41	50	50	44	56	44	38
PE – Construction I	6	12	18	12	23	23	35	38	35
PE – Construction II	3	9	18	15	23	23	35	44	44

Figure 8: Experiment results for acoustic models trained **excluding** 'explicit noise modelling'. Language model grammar is **activated** in this configuration.

as it can be observed from figure 9, the accuracy ratio for 'PL - Vacuum-cleaner' sample in AM00 was 50 per cent raised, by utilising Audacity 'noise removal' feature. Whereas, for 'PE - Vacuum-cleaner' sample in the same AM, accuracy ratio was lowered more than 60 per cent.

Prior to performing the experiment, we filtered a few samples provided by Volvo Technology. The samples were recorded in the same noisy environment that the final application is going to be deployed at. We used Audacity to retrieve the sample's voice spectrum. With the spectrum view, we were able to select the noise spectrum manually. By calculating the Fast Fourier Transform on the selected noise spectrum, we discovered the cepstrum frequency in time-domain. We removed the high decibel frequency of noise cepstrum, by using STK framework Notch filter. This technique resulted in improving accuracy ratio significantly, as we expected.

For our experiment, we did not have the opportunity to manually select the noise spectrum from each recorded audio file. Thus, we considered the first five milliseconds of each audio file as background noise, approximately one second before the command was pronounced. Subsequently, we considered the highest decibel frequency after 0.01 milliseconds as additive noise. We suspected the high decibel frequencies before 0.01 milliseconds were caused by channel distortion and not additive noises. Afterwards, we cut the selected noise frequency by utilising Notch filter of STK framework.

As it is demonstrated in figure 12, Notch filter was not capable of increasing the accuracy ratio in non-stationary noisy environment. Our results show, Notch filter in the best scenario did not decrease the accuracy ratio for non-stationary noisy environments. This means, no improvement was made by Notch filter in any scenario. In contrast to that, applying the same technique on stationary vacuum-cleaner noise produced slightly satisfactory results. As it can be observed from figure 11, Notch filter raised the accuracy ratio of 'PE - Vacuum-cleaner' sample up to 40 per cent in AM02 and AM04. In the same AMs, accuracy ratio of 'PL - Vacuum-cleaner' sample was also improved by Notch filter.

In addition to STK framework, we filtered the same samples by Audacity 'noise removal' feature, which is a pattern recognition noise cancellation. In this process, we selected the noise profile manually from one of the recorded files in each sample. Then, we set 'noise reduction level' and 'frequency smoothing' to 48-db and 0 Hz respectively. We used the default value of 0.15 for 'decay time'. The influence of this technique on stationary noise was slightly better.

As it is illustrated in figure 9, Audacity 'noise removal' feature boosted the accuracy ratio for the sample recorded with Plantronics microphone, in most of the AMs – except AM07 and AM08. While for the same sample recorded with Peltor microphone, improvement can only be noticed in AM02. Figure 10 shows the influence of Audacity 'noise removal' feature on Construction II sample – non-stationary noise. For the 'PL - Construction II' sample, only in AM05 less than 10 per cent improvement occurred. The same sample recorded with the other microphone made slightly more improvement.

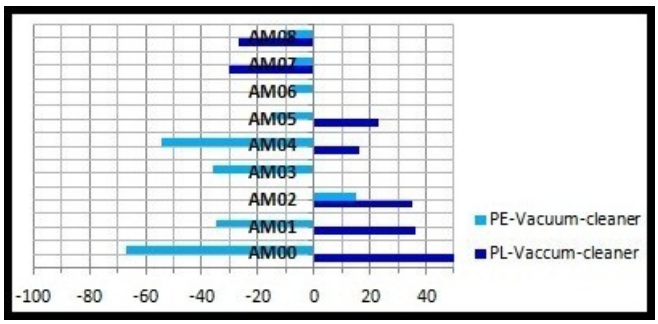


Figure 9: Influence of **Audacity 'noise removal' feature** on accuracy ratio of vacuum-cleaner sample.

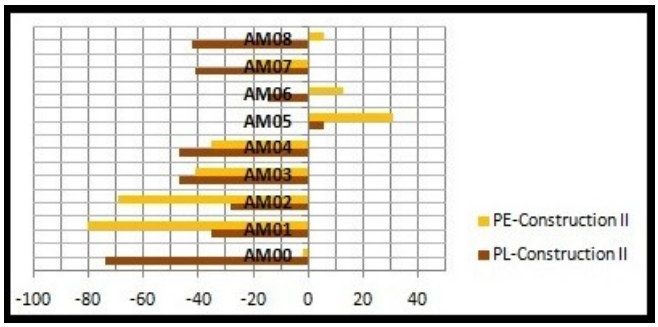


Figure 10: Influence of **Audacity 'noise removal' feature** on accuracy ratio of construction II sample.

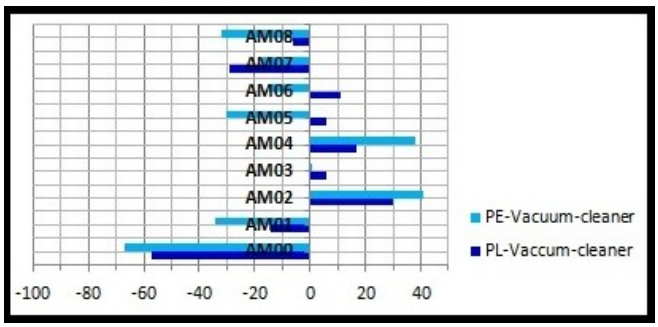


Figure 11: Influence of **Notch filter** on accuracy ratio of vacuum-cleaner sample.

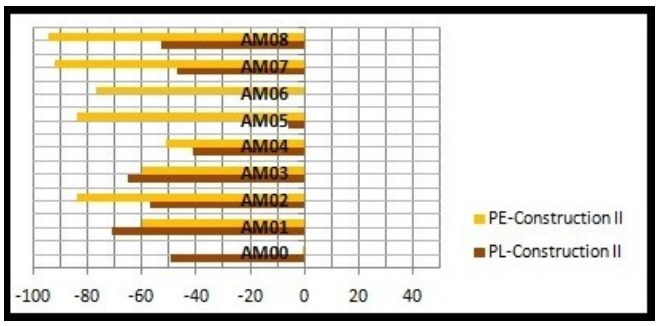


Figure 12: Influence of **Notch filter** on accuracy ratio of construction II sample.

iii Language model

We created our language model grammar in Java Speech Grammar Format (JSGF). Refer to Appendix D for detailed grammar file. Consequently, we examined all the recorded samples by activating the grammar. For instance, in our prototype "BUCKET UP FIVE DEGREES" is grammatically correct, whereas "BUCKET FIVE DEGREES UP" is incorrect.

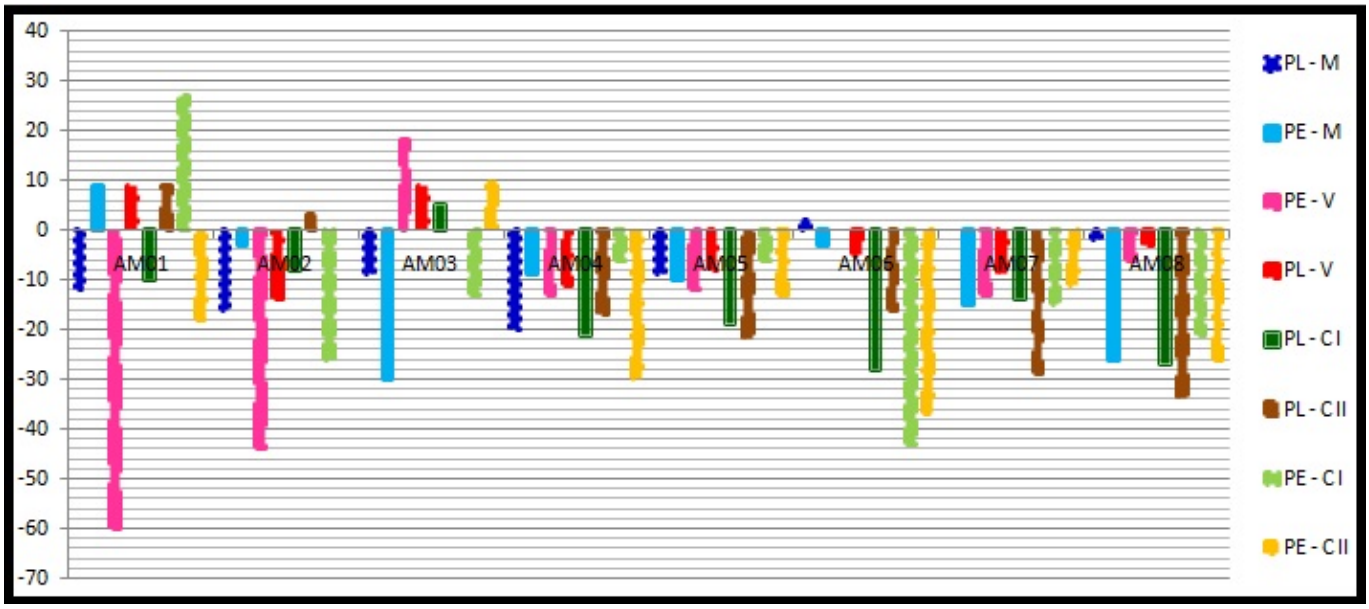


Figure 13: The influence of grammar mode on accuracy ratio of recognised commands.

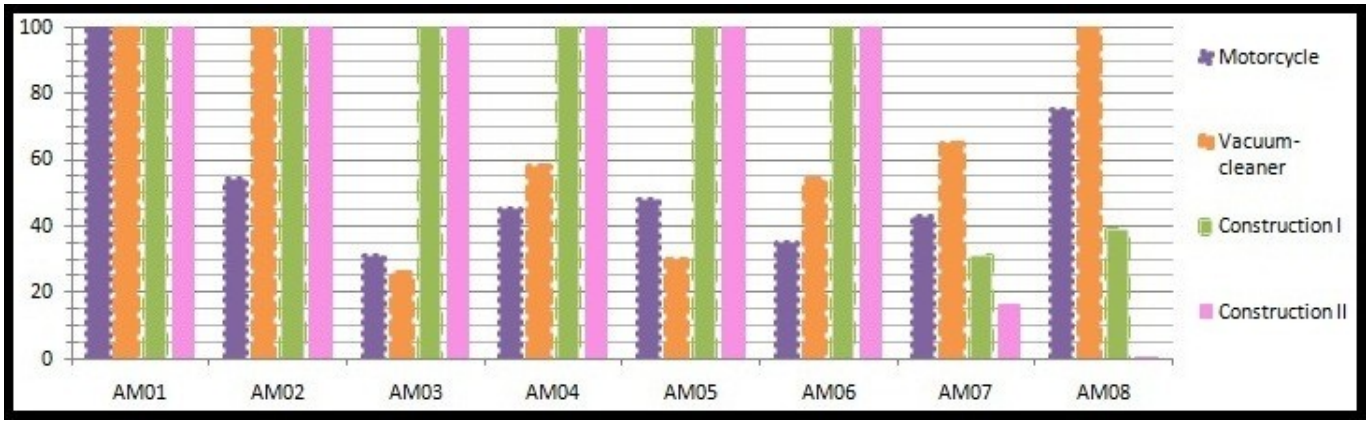


Figure 14: This figure illustrates average percentage improvement Plantronics microphone produced, comparing to the Peltor one.

Therefore, the later command would be rejected by the decoder.

Next, we compared the experiment results from the grammar inactivated mode – figures 5 and 6 – with the results of experiments from grammar activated mode – figures 7 and 8. The comparison shows that including natural language grammar degrades the accuracy ratio. The reason behind this is that the application restricts itself to grammar, making the recognition not as accurate as expected. For instance as it can be observed in figure 6, even though in grammar inactivated mode 68 per cent of the commands in 'PE - Construction II' environment were recognised correctly; only 44 per cent of commands were recognised correctly in grammar activated mode, as it can be noticed in figure 8.

In figure 13, for each sample we illustrate the average percentage of accuracy ratio that was lowered due to having grammar activated. For example, in AM01 for 'PE - Vacuum-cleaner' about 60 per cent accuracy ratio was declined, when grammar was activated. In very rare cases – 9 out of 64 – accuracy ratio was improved in grammar activated mode.

iv Microphone characteristics

We compared the two different microphones, which we used for our recording samples. The results show that the Plantronics microphone produced superior accuracy ratio comparing to the Peltor one. Figure 15 shows the amount of improvement for each AM when Plantronics microphone was used. For instance, in AM02 under motorcycle environment, the Plantronics microphone improved the accuracy ratio 50 per cent. Whereas, for the other three environments, the improvement was over 100 per cent. Although, we cannot explain the reason behind it, we can argue that in order to have a robust SR application, a right microphone must be selected.

4 Ultimate experiment

To finalise our experiment, we travelled to Eskilstuna to examine our prototype in construction settings of Volvo Construction Equipment. We recorded five different samples with each microphone in a demo environment, while a

wheel loader was working on loading and dumping stones. Same as our primary experiment, all recordings were single-channel – monaural, little-endian, unheadered 16-bit signed PCM audio file sampled at 16000 Hz. And all the collected audio samples had a unique speaker. The noise level was approximately 80-db at the most.

The speaker's position while recording the commands was as follow:

- The first sample in-front of the machine.
- The second and third samples inside the cabin while radio was off and on respectively.
- The fourth and fifth samples on left and right side of the machine respectively.

For each microphone, we trained the primary AM in the following steps, while 'explicit noise modelling' was excluded:

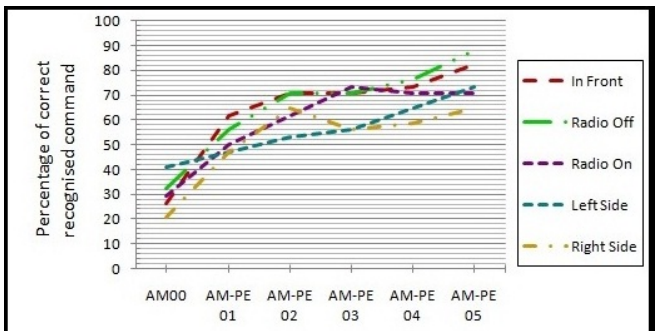
1. Training the AM00 with the first sample, as a consequence AM01 was created.
2. Training the AM01 with the second sample, as a consequence AM02 was created.
3. Training the AM02 with the third sample, as a consequence AM03 was created.
4. Training the AM03 with the fourth sample, as a consequence AM04 was created.
5. Training the AM04 with the fifth sample, as a consequence AM05 was created.

Consequently, we produced ten new AMs – AM-PL01 to AM-PL05 for the Plantronics microphone and AM-PE01 to AM-PE05 for the Peltor one.

Thereafter, we examined the samples recorded with Plantronics microphone with its own trained AMs, and the samples recorded with Peltor microphone with its own trained AMs. It must be mentioned, that we inactivated the LM grammar for our ultimate experiment. As it can be observed in figures 15 and 16, the ultimate experiment results show a significant improvement in all the trained AMs.

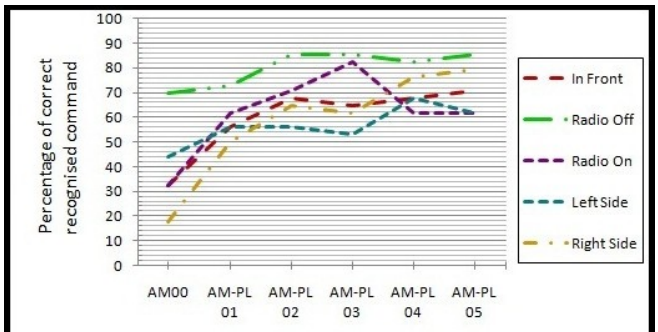
In figure 15, the accuracy ratios of all trained AMs are almost twice as the untrained one – AM00. For instance, for the sample recorded with Peltor microphone inside the cabin, while radio was off, AM00 correctly recognised only 32 per cent of all commands. Whereas, in AM-PE-05 accuracy ratio was 88 per cent.

Similarly for the Plantronics microphone, the accuracy ratios of all trained AMs are higher than the untrained one – AM00. For example, as it can be observed from figure 16, for the sample recorded on the right side of wheel loader, AM00 recognised only 18 per cent of all commands correctly. Whereas, in AM-PL-05 accuracy ratio was boosted to 80 per cent.



	AM 00	AM-PE 01	AM-PE 02	AM-PE 03	AM-PE 04	AM-PE 05
In Front of Machine	27	62	70	70	73	82
Inside Cabin-Radio Off	32	56	70	70	76	88
Inside Cabin-Radio On	29	50	62	73	70	70
Left Side of Machine	41	47	53	56	65	73
Right Side of Machine	20	47	65	56	59	65

Figure 15: Ultimate experiment for samples recorded with the **Peltor** microphone. Acoustic models were trained excluding 'explicit noise modelling'. Language model grammar was inactivated.



	AM 00	AM-PL 01	AM-PL 02	AM-PL 03	AM-PL 04	AM-PL 05
In Front of Machine	32	56	68	65	68	70
Inside Cabin-Radio Off	70	73	85	85	82	85
Inside Cabin-Radio On	32	62	70	82	62	62
Left Side of Machine	44	56	56	53	68	61
Right Side of Machine	18	50	65	62	76	80

Figure 16: Ultimate experiment for samples recorded with the **Plantronics** microphone. Acoustic models were trained excluding 'explicit noise modelling'. Language model grammar was inactivated.

5 Discussion

This section is dived into four subsections – i.e. (i) acoustic model, (ii) speech quality, (iii) language model, and (iv) microphone characteristics. In each subsection, we discuss our findings from the results of experiments, and map them onto the solutions, explained in the background section.

1 Acoustic model

As we illustrated in the experiment section, AMT can significantly improve accuracy ratio of ASR applications in heavy noisy environments. Refer to the four figures – 5, 6, 7, and 8 – of our primary experiment and two figures – 15 and 16 – of our ultimate experiment for further details.

However, as it can be observed from the results of our experiment, accuracy ratio never reached the perfect percentage. This implies that there is still room for improvement. There are different means to improve capabilities of AM, which naturally boosts noise robustness. Implementing those means require further research. In the following subsections, we discuss three of those means.

i Building acoustic model from scratch

It is possible to improve the accuracy ratio of ASR applications by AMT; however it is better in many cases to build the AM from scratch depending on domain requirements. According to Humphries et al. [14], SR engines work best when AM is trained with speech audio that was recorded at the same sampling rate or bits per sample as the speech being recognised. Building a new AM requires intensive work for months, which was not feasible within the time-span of our research.

In CMU-Sphinx AMT tutorial[10], it is advised to build AM in the following circumstances.

- It is required to create an AM for new language or dialect.
- Specialised model is required for small vocabulary application.
- Following data are available:
  - 1 hour of recording for command and control for single speaker.
  - 5 hours of recordings of 200 speakers for command and control for many speakers.
  - 10 hours of recordings for single speaker dictation.
  - 50 hours of recordings of 200 speakers for many speakers dictation.
- Sufficient knowledge on phonetic structure of the language is available.
- There is time to train the model and optimise parameters – one month.

And it is recommended to train AM in the following circumstances.

- The aim is to improve accuracy – perform AMT instead.
- Not enough data is available – perform AMT instead.
- There is time constraint.
- There is lack of experience.

ii More training required

For the ultimate experiment, we trained the AM with five different samples, as it was explained in its corresponding section. The improvement was significant, specifically for the last trained AM – AM05. The training process must be more extensive with larger recorded samples for industrial applications. To achieve this, Huang et al. [13] suggest vocabulary-dependent (VD) training on a large population of speakers for each vocabulary. However, these training demands months for data collection, weeks for dictionary generation, and days for data processing.

iii More precise 'explicit noise modelling'

Firstly, including 'explicit noise modelling' in AMT requires great number of filler words. In our prototype, we only had eight filler words, such as ++NOISE++, ++BREATH++, ++UM++, and ++SMACK++. These words did not represent all the different existing background noises in our recorded samples. Due to this fact, during the training we mapped ++NOISE++ onto many different types of noise – e.g. wind and engine sound. This might be the reason why 'explicit noise modelling' did not improve the accuracy ratio of AMT in our research.

Secondly, 'explicit noise modelling' must be performed with patience. This means, all the recorded samples must be carefully analysed, and the transcription must be accordingly changed. 'Explicit noise modelling' is strongly recommended by different literature [12] [22]. Hence, we believe if there had been more filler words like ++WIND++ and ++STONE++, the mapping would have been more precise and the 'explicit noise modelling' could have improved the accuracy ratio.

2 Speech quality

As it was mentioned in the background section, removing noisy features from input signals increases accuracy ratio of SR systems. Gillian [11] explains, if noise and speech do not share the same frequency range, digital filtering is a promising technique. On the other hand, the task becomes cumbersome when noise and speech overlap in frequency.

Our experiment results showed the accuracy ratio can be improved by using filters. However, it requires advanced signal processing knowledge; due to the fact that the noises we can hear are in the same frequency range of human voice. In the following two subsections, we discuss our findings from the experiments and map them onto the techniques described in the background section.

i Stationary noise removal

Stationary noise features can be removed from signal by using Elliptic Notch filter, as explained in the background section. We chose vacuum-cleaner sample, which is categorised as stationary noise. As suggested by Gillian [11], we processed our signal in a transform domain – Fourier Transform – and tried to filter the background noise. Results were not promising. We suspected not perfectly identifying noise characteristics could be the reason why Notch filter was unfruitful for our project.

As it can be observed from figure 11, in more than half of the AMs, Notch filter even decreased the accuracy ratio. However, in some cases the accuracy ratio was improved. As an instance, in AM02 and AM04 which were trained by stationary background noise – motorcycle and vacuum-cleaner respectively – we observed approximate 40 per cent improvement. This implies, for stationary background noise Notch filter could be effective.

[10] http://cmusphinx.sourceforge.net/wiki/tutorialadapt

Moreover, we tried to remove the background noise of vacuum-cleaner by using Audacity 'noise removal' feature. Figure 9 illustrates the efficiency of the 'noise removal'. From figures 11 and 9, we can conclude that it is possible to reduce influence of stationary noise from signal, yet advanced signal processing knowledge is required to address the noise characteristic.

ii Non-stationary noise removal

Non-stationary noise characteristic varies overtime; it is impossible to reduce its impact by using Notch filter which only blocks one or two frequency bands. As it can be seen in figure 12, Notch filter was not efficient for reducing the influence of non-stationary noise. Gillian [11] recommends using adaptive algorithms when noise is periodic. Noise cancelation is one of the most competent techniques for removing this type of noise – as introduced in the background section.

We tried to remove the construction setting background noise from input signal by using Audacity 'noise removal' feature. This feature has pattern recognition algorithm. We tried to manually select the noise pattern, so the system can recognise it. However, the results were not hopeful. We suspect not precisely decide on the noise pattern and its behaviour can be the reason for our findings. Refer to figure 10 for more details.

3 Language model

Kuhn et al. [16] state that ASR generally consists of two components. (i) An acoustic component that matches the acoustic input to words in its vocabulary, producing a set of the most plausible word candidates together with a probability of each. And (ii) an LM, which estimates for each word in the vocabulary the portability that it will occur, given a list of previously hypothesised words. This shows importance of LM for ASR applications. In the following two subsections, we study two different elements of LM: (i) grammar, and (ii) dictionary.

i Grammar

As it is illustrated in figure 13, including grammar in noisy environments can actually decline accuracy ratio of application. This is due to the fact that decoder is forced to map any type of phoneme onto its grammar. Subsequently, it suggests false hypothesis.

We reached the proposal of activating grammar after the decoder formulates hypotheses. This means, after retrieving the hypothesis from the decoder, the highest hypothesis that matches the grammar is selected. Due to time constraints, we did not implement this proposal.

ii Dictionary

As it was mentioned in the background section, SR is fundamentally a pattern-matching problem. Thus, if the way speaker pronounces a word is not included in the dictionary file, the decoder will not be able to match that word to any words in its dictionary.

One of the methods to overcome this barrier is to add new alternative pronunciations in the dictionary file. This is specifically helpful for non-native speakers. For instance, if the default pronunciation for 'hundred' is 'HH AH N D R AH D', but a speaker pronounces it as 'HH AH N D R IH D', the decoder will not be able to recognise the speaker's pronunciation. However, by adding the second pronunciation as an alternative, the application can recognise the speaker's pronunciation as well. It must be mentioned that there is no limitation on the number of alternative pronunciations for a word, i.e. a word can have as many different pronunciation as it is required.

4 Microphone characteristics

As it is illustrated in figure 14, different microphones can generate diverse results. We cannot explain why one microphone produced better results than the other one. This could be caused by variety of reasons, for instance distortion, electrical noise, or directional characteristics. Regardless, it is certain that choosing a correct microphone and calibrating it properly improves the noise robustness significantly.

6 Conclusion

In this study we illustrated whether SR is feasible in heavy noisy environments, specifically in construction settings. We initially presented current state-of-the-art and state-of-the-practice. Subsequently, we showed the influence of four different elements on noise robustness, namely (i) acoustic model, (ii) speech quality, (iii) language model, and (iv) microphone characteristics.

To summarise, we showed that AMT is indispensable for reaching a noise robust application. We also illustrated that it is important to train AM in the same environment that application is going to be deployed at, and record the samples with the same microphone that is intended to be used for the final application. Subsequently, we showed noise reduction and filtering techniques can boost the accuracy ratio of SR applications, however deeper investigations are required.

We also examined LM, i.e. word dictionary and grammar. The results demonstrated that grammar is not efficient for VC in noisy environments. Last but not least, we showed that microphone is an influential element in SR. Thus, it is important to choose a correct microphone for noisy environments.

Finally, we believe the list presented below can be the potential future study of our research:

- Acoustic model
  - Focusing on 'explicit noise modelling' by having more filler words, and mapping precisely different noises onto their corresponding filler words, in order to show 'explicit noise modelling' improves AMT.
  - Building a special AM for heavy noisy construction settings.

- Speech quality
  - Having the option for calibrating the system before the user pronounces the commands. In order to recognise noise characteristic and distinguish them completely from speech.
  - Implementing the adaptive filter for the desired environment, so the filter adapts itself to the noise of that environment.
- Language model
  - Implementing grammar after the speech decoder formulates hypotheses, to examine whether that improves accuracy ratio.
- Microphone characteristics
  - Selecting a microphone array over the conventional directional microphones. This can improve SNR, response for arbitrary speaker position, and speech period detection in noisy environment.

7 Acknowledgement

Conducting this bachelor thesis was indeed a rewarding experience. We would like to thank Volvo Technology and Volvo Construction Equipment, specially Torbjörn Martinsson, Stefan Bergquist, and Filip Holmqvist for all their supports and guidance with providing ideas and feedback during this research. Great thanks to IT University of Gothenburg and its professors, especially Gerardo Schneider for all the ideas and supports that makes this research possible to be presented to you.

References

[1] BASILI, V., SHULL, F., AND LANUBILE, F. Building knowledge through families of experiments. *Software Engineering, IEEE Transactions on 25*, 4 (1999), 456–473. 3

[2] BASILI, V. R., SELBY, R. W., AND HUTCHENS, D. H. Experimentation in software engineering. *IEEE Trans. Softw. Eng. 12*, 7 (1986), 733–743. 4

[3] BECCHETTI, C., AND RICOTTIL, L. P. *Speech Recognition: Theory and C++ Implementation*. Chichester: Wiley, 1999. 1, 2

[4] BENESTY, J., SONDHI, M. M., AND HUANG, Y. A. *Springer Handbook of Speech Processing*. New York: Springer, 2007. 1, 3, 5, 6

[5] BOOTH, W. C., COLOMB, G. G., AND WILLIAMS, J. M. *The Craft of Research*, second ed. Chicago: University of Chicago Press, 2003. 2, 3

[6] BRUSAW, C. T., ALRED, G. J., AND OLIU, W. E. *Handbook of Technical Writing*. Fifth Edition. New York: St. Martin's Press., 1997. 3

[7] CAMPBELL, D., AND STANLEY, J. *Experimental and Quasi-Experimental Design for Research*. Rand McNally., 1966. 3

[8] COOK, T., AND CAMPBELL, D. *Quasi-Experimentation: Design and Analysis for Field Settings*. Rand McNally., 1979. 4

[9] CRESWELL, J. W. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, second ed. London: Sage Publications, 2002. 4

[10] GALVAN, J. *Writing Literature Reviews*. Pyrczak Publishing., 1999. 3

[11] GILLIAN, D. *Noise Reduction in Speech Applications*. CRC Press, Inc., Boca Raton, FL, USA, 2002. 6, 14, 15

[12] HUANG, X., ACERO, A., AND HON, H. W. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. New Jersy: Prentice Hall PTR, 2001. 4, 5, 6, 7, 14

[13] HUANG, X., ALLEVA, F., WUEN HON, H., YUH HWANG, M., AND ROSENFELD, R. The sphinx-ii speech recognition system: An overview. *Computer, Speech and Language 7* (1992), 137–148. 14

[14] HUMPHRIES, J., AND WOODLAND, P. The use of accent-specific pronunciation dictionaries in acoustic model training. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on* (may 1998), vol. 1, pp. 317 –320 vol.1. 14

[15] ILIEV, G., AND KASABOV, N. Adaptive blind noise suppression in some speech processing applications. *Software Engineering, IEEE Transactions* (1999). 6

[16] KUHN, T. *The Structure of Scientific Revolutions*. University of Chicago Press., 1970. 2, 15

[17] LAKATOS, I. *Criticism and the Methodology of Scientific Research Programs*. Proceedings of the Aristoellan Society., 1968. 2

[18] PIERCE, D., AND GUNAWARDANA, A. Aurora 2.0 speech recognition in noise. *Proc. Speech and Natural Language Workshop* (2002), 311–318. 5

[19] RABINER, L., AND JUANG, B. H. An introduction to hidden markov models. *IEEE Assp Magazine 3* (1986), 4–16. 5

[20] SADAOKI, F. Toward robust speech recognition and understanding. *The Journal of VLSI Signal Processing 41* (2005), 245–254. 7

[21] VAISHNAVI, V., AND KUECHLER, W. Design research in information systems. http://desrist.org/design-research-in-information-systems, 2004. Last updated August 16, 2009. 1, 2, 3

[22] WARD, W. Modelling non-verbal sounds for speech recognition. Association for Computational Linguistics, pp. 47–50. 6, 14

[23] WIKIPEDIA. Filter (signal processing). http://en.wikipedia.org/wiki/Filter_%28signal_processing%29, 2011. Accessed May 22, 2011. 6

[24] ZORNETZER, S. F., DAVIS, J. L., AND LAU, C. *An introduction to neural and electronic networks*, second ed. Academic Press Professional, Inc., 1990. 6

Appendices

A Glossary

AM	Acoustic Model
AMT	Acoustic Model Training
ASR	Automatic Speech Recognition
CFG	Context-Free-Grammar
db	Decibel
EM	Expectation-Maximisation
HMM	Hidden Markov Model
Hz	Hertz
JSGF	Java Speech Grammar Format
LM	Language Model
PCM	Pulse-Code Modulation
SNR	Signal-to-Noise Ratio
SR	Speech Recognition
TDD	Test Driven Development
VC	Voice Command
VD	Vocabulary Dependent

B Experiment

We conducted a simple SR experiment on two existing applications – Android 'Speech Input', and Windows 7 'Speech Recognition'. The experiment was performed with two different devices – i.e. an HTC mobile device[11] and a Lenovo laptop computer[12]. The experiment circumstances were as follow:

- A unique speaker on both devices.
- Both devices were tested under exactly the same environments – i.e. without any background noise in a quiet room and with the background noise, featuring construction sound.
- We used Plantronics microphone for the Windows 7 experiment, whereas for the Android experiment, we used an iPhone Stereo Headset. We did not use the same microphone for both devices, because there was no easy way to plug the Plantronics microphone into the HTC device.

The result of the experiment is described in table 3. The first column is the commands we pronounced. The second and third columns show the results for Android 'Speech Input' in quiet and noisy environments respectively. Similarly, the fourth and fifth columns show the results for Windows 7 'Speech Recognition'.

[11] Android Froyo, 1 GHz Snapdragon CPU, and 512 MB RAM
[12] Windows 7, Triple-Core 2.10 GHz CPU, and 4,00 GB RAM

Command	Android		Windows 7	
	Quiet	Noisy	Quiet	Noisy
Bucket up	✓	X	✓	X
Tilt out	✓	X	✓	X
Lift down	X	X	✓	X
356 degrees	✓	✓	✓	✓
Waist in 9 centimetre	X	X	✓	X
Stop listening	✓	✓	✓	✓
Safe mode	✓	X	✓	✓
Tilt left 18 centimetre	X	X	X	X
Parallel mode	✓	X	X	X
Andrew start	✓	✓	X	X

Table 3: Experiment outcome for **Android** 'Speech Input' and **Windows 7** 'Speech Recognition'.

C Commands

Below is listed the commands that are used for the controlling of construction machines, such as wheel loader and excavator.

- Function controls
  - Lift up | down
  - Lift up | down xx degrees (differentiate)
  - Tilt in | out
  - Tilt in | out xx degrees (differentiate)
  - Waist right | left
  - RPM xx
- Body controls
  - Forward | backwards
  - Forward | backwards xx cm (differentiate)
  - Bucket right | left
  - Bucket right | left xx cm (differentiate)
  - Bucket up | down
  - Bucket up | down xx cm (differentiate)
  - Lock | Unlock position
  - Stop
- Set-up controls
  - Slow mode
  - Normal mode
  - Safe mode
  - Parallel mode
  - Bucket tip mode
- Miscellaneous
  - Stop listening
  - Start Listening
  - Safir

D Grammar

```
#JSGF V1.0;

/**
 * JSGF Grammar for Commands
 */

grammar safir.command;

public <all> = [ SAFIR ] ( <actions> | <modes> | ( <commands> <directions> ) [ ( <one_digitis> | <two_digitis> | <three_digitis> )↩
      <scales> ] );

<actions> = ( STOP | START ) [ LISTENING ] | LOCK | UNLOCK;

<modes> = (SLOW | NORMAL | SAFE | PARALLEL | BUCKET TIP) (MODE);

<commands> = BUCKET | TILT | LIFT | WAIST | FORWARD | BACKWARD;

<directions> = FORWARD | BACKWARD | UP | DOWN | RIGHT | LEFT | IN | OUT;

<one_digitis> = ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN | EIGHT | NINE;
<two_digitis> = TEN | ELVEN | TWELVE | THIRTEEN | FOURTEEN | FIFTEEN | SIXTEEN | SEVENTEEN | EIGHTEEN | NINETEEN | <twenties>;
<twenties> = ( TWENTY | THIRTY | FORTY | FIFTY | SIXTY | SEVENTY | EIGHTY | NINETY ) [ <one_digitis> ];
<three_digitis> = ( <one_digitis> HUNDRED ) [ <two_digitis> | <one_digitis> ];

<scales> = CENTIMETER | DEGREES;
```