Automatic speech recognition

An evaluation of Google Speech

Magnus Stenman

Magnus Stenman
Spring 2015
Bachelor’s thesis, 15 hp
Supervisor: Marie Nordström
Examiner: Jerry Eriksson
Bachelor’s programme in Computing Science, 180 hp

Abstract

The use of speech recognition is increasing rapidly and is now available in smart TVs, desktop computers, every new smart phone, etc. allowing us to talk to computers naturally. With the use in home appliances, education and even in surgical procedures accuracy and speed becomes very important.

This thesis aims to give an introduction to speech recognition and discuss its use in robotics. An evaluation of Google Speech, using Google's speech API, in regards to word error rate and translation speed, as well as a comparison between Google Speech and Pocketsphinx is made.

The results show that Google Speech presented lower error rates on general purpose sentences but, due to the high average translation speed and the inability to specify vocabulary, was not suitable for voice-controlled moving robots.

Acknowledgements

I would like to thank the people at Umeå University who has helped out during this work:

Thomas Hellström
for providing the idea and an introduction to the field of speech recognition

Peter Hohnloser
for the excellent support regarding ROS and AmigoBot

Marie Nordström
for providing advice the whole way through, helping me write a better paper

Rickard Hjulström
for working together with the data collections and coding, making the comparisons possible

Contents

1 Introduction 1

1.1 Purpose 1

1.2 Voice-controlled robot 1

2 An introduction to speech recognition 3

2.1 The sounds of speech 3

2.2 Accuracy of automatic speech recognition 3

2.3 The speech recognition process 4

2.4 Hidden Markov models 6

2.5 Gaussian mixture models 7

2.6 Neural networks 7

2.7 Google Speech 8

3 Methods of measuring 9

3.1 Data collection 9

3.2 Implementation and calculation 9

3.3 Word error rate 10

4 Results 11

4.1 Word error rate comparison 11

5 Controlling a robot with speech recognition 13

5.1 The Robot Operating System (ROS) 13

5.2 Speech recognition in robotics 14

5.3 Implementation and testing 14

5.4 Robot commands explained 15

5.5 Data collection and results 16

6 Conclusions 17

7 Discussion and Future work **19**

References **21**

A Recorded sentences **25**

A.1 Recorded command sentences 27

B Setup specification **29**

1 Introduction

The definition of speech recognition according to Macmillan Dictionary [1] is *”a system where you speak to a computer to make it do things, for example instead of using a keyboard”*. While the definition is true, as the area of artificial intelligence moves forward the applications for speech recognition has rocketed. In smart TVs, desktop computers, every new smart phone, etc. we now have access to voice control through speech recognition. There have also been successful uses of voice controlled systems in both medicine [2] [3] and in education for blind and/or otherwise handicapped people [4].

To be able to communicate with devices in a natural way, we need speech recognition. This, of course, makes it necessary to have great accuracy, fast speed and the ability to recognize many different speakers. With today’s internet availability, Google Speech can be used for getting fast and accurate results as Google has the capability and database to translate spoken language to text using their own servers processing power (cloud-based computing).

The problems that can arise with speech recognition include background noise interference, differences in accents, dialects and physiology affecting our speech, vocabulary size and content, detecting utterances such as coughs as non-words, etc.

1.1 Purpose

In this thesis an evaluation of Google Speech will be made using recordings in English from two Swedish speakers based on word error rate (WER) and translation speed. The evaluation will also include audio files with artificially added background noise as well as different sentence lengths. The results will then be compared to another speech recognition system, Pocketsphinx, using the same recordings and measurements made in a separate bachelor thesis titled ”Evaluation of a speech recognition system: Pocketsphinx” by Rickard Hjulström.

The questions that will be discussed in this thesis are as follows:

- How does Google Speech compare to another speech recognition system regarding word error rate and translation speed?
- Which system is best suited for controlling a robot using voice commands?

1.2 Voice-controlled robot

The use of controlling a robot with voice commands using speech recognition will also be reviewed. This part will include a proof of concept using the AmigoBot [5], which has been fitted with a wireless card and connects to a Robot Operating System (ROS) server. Another evaluation using the same principles will be made, now using recordings of robot commands as input data and a modified vocabulary.

2 An introduction to speech recognition

A speech recognition (SR) system can basically be either speaker-dependent or speaker-independent. A speaker-dependent system is intended to be used by a single speaker and is therefore trained to understand one particular speech pattern. A speaker-independent system is intended for use by any speaker and is naturally more difficult to achieve. These systems tend to have 3 to 5 times higher error rates than speaker-dependent systems [6].

2.1 The sounds of speech

To understand SR, one should understand the components of human speech. A phoneme is defined as the smallest unit of speech that distinguishes a meaning [7], e.g. the word ”speech” has the four phonemes: S P IY CH. Every language has a set number of phonemes which will sound different depending on accents, dialects and physiology [8].

When phonemes are considered in SR, they can be considered in their acoustic context, making them sound different, i.e. when also considering the phoneme to the left or right of the phoneme we’re trying to interpret we call them biphones. When considering both left and right context we call them triphones [7].

Continuous speech is complicated because when we speak, as a particular articulatory gesture is being produced the next one is already being anticipated and therefor changing the sound. This phenomenon is called co-articulation, the smearing of sounds into one another. Human speech also have variations in pitch, rhythm and intensity, e.g. we stress certain words to get our meaning through [8].

2.2 Accuracy of automatic speech recognition

The accuracy of an SR system is commonly measured with WER [7] [9], and that is also the unit used in this thesis.

$$WER = \frac{Number\ of\ Substitutions + Insertions + Deletions}{Total\ number\ of\ words} \tag{2.1}$$

However, the conditions of evaluation and therefore the accuracy of the system can vary in a number of areas [6]:

- Vocabulary size and confusable words
  With a small vocabulary, it’s easier for the system to recognize the correct word compared to a larger one. Error rates naturally increases with the vocabulary size. For example the numbers zero through ten can be recognized essentially perfectly, but

with increased vocabulary sizes or the addition of confusable words, i.e. words that sound alike, the error rates increases. For example the words *dew* and *you* is very similar in sound, but not at all in meaning.

- Speaker dependence vs. independence
  A speaker-dependent system, depending on training and speaker, is usually more accurate than the speaker-independent system. There are also multi-speaker systems that are intended to be used by a small group of people and speaker-adaptive systems that learn to understand any speaker given a small amount of speech data for training.

- Isolated, discontinuous, or continuous speech
  Isolated, meaning single words, and discontinuous, meaning full sentences with artificially separated words by silence, are the easiest to recognize since the boundaries are detectable. Continuous speech is the most difficult one to recognize because of co-articulation and unclear boundaries, but it's the most interesting one since it allows us to speak naturally.

- Task and language constraints
  The constraints can be task-dependent, accepting only relevant sentences for the task, e.g. an ticket purchase service rejecting ”The car is blue”. Others can be semantic, rejecting ”The car is sad” or syntactic, rejecting ”Car sad the is”. Constraints are represented by grammar, filtering out unreasonable sentences and is measured with their perplexity, a number representing the grammars branching factor, i.e. the number of words that can follow a specific word.

- Spontaneous vs. read speech
  Read speech from a text is easy to understand compared to spontaneous speech where words like ”uh” and ”um”, stuttering, coughing and laughter can occur.

- Recording conditions
  Performance is affected by background noise, acoustics (such as echoes), type of microphone (e.g. close-speaking, telephone or omnidirectional), limited frequency bandwidth (for example telephone transmissions) and altering speaking manners (shouting, speaking quickly, etc.).

2.3 The speech recognition process

The common method used in automatic speech recognition systems is the probabilistic approach, computing a score for matching spoken words with a speech signal. A speech signal corresponds to any word or sequence of words in the vocabulary with a probability value [7]. The score is calculated from phonemes in the acoustic model knowing which words can follow other words through linguistic knowledge. The word sequence with the highest score gets chosen as the recognition result.

The SR process can be divided into four consecutive steps; pre-processing, feature extraction, decoding and post-processing. Different SR systems have different implementations of each step and in between them, the following is just an example.

Pre-processing is the recording of speech with a sampling frequency of, for example, 16 kHz and, according to The Shannon Theorem [10], a bandwidth limited signal can be reconstructed if the sampling frequency is more than double the maximum frequency meaning that frequencies up to almost 8 kHz are constituted correctly. It has been shown that data transmitted over telephone network, ranging from 5 Hz to 3.7 kHz, is sufficient for recognition so 8 kHz is more than enough. All frequencies below 100 Hz can be removed as they are considered noise. One important part of pre-processing is to remove the parts between the recording starts and the user starts talking as well as after the end of speech. This is done to counter the fact that a SR system will assign a probability, even if very low, to any sound-phoneme combination making background noise insert phonemes into the recognition process [7].

Speech signals are slowly timed varying signals and their characteristics are fairly stationary when examined over a short period of time (5-100 ms) [11]. Therefore, in the **feature extraction** step, acoustic observations are extracted in frames of typically 25 ms. For the acoustic samples in that frame, a multi-dimensional vector is calculated and on that vector a fast Fourier transformation is performed [7] [11], to transform a function of time, e.g. a signal in this case, into their frequencies. A common feature extraction step is cepstral mean subtraction (CMS) which is used to normalize differences between channels, microphones and speakers [7] [12].

In the **decoding** process is where calculations is made to find which sequence of words that is the most probable match to the feature vectors. For this step to work, three things has to be present; an acoustic model with an hidden Markov model (HMM) for each unit (phoneme or word), a dictionary containing possible words and their phoneme sequences and a language model with words or word sequences likelihoods [7].

An example of a dictionary entry is ”recognition R EH K AH G N IH SH AH N”, the word followed by its phonemes. The language model is typically fixed grammars, or n-gram models, with a 1-gram (unigram) model only listing words and their probability and a 2-gram (bigram) model listing words and their probability when followed by some other word and so on [7]. An example of a 1-gram, a 2-gram and a 3-gram (trigram) model created from the sentence ”this is an example” is listed below.

- Possible 1-gram sentences:

  this

  is

  an

  example

- 2-gram:

  this is

  is an

  an example

- 3-gram:

  this is an

  is an example

Imagine n-gram sentences, with their calculated probabilities, created from one or multiple much larger texts, called a corpus or corpora. This creates a language model with correct grammar, given that proper sources were used.

In decoding the SR systems tries to find the word or sequence of words w* best matching the observation X, giving the equation 2.2 with $p(w)$ being from the language model and $p(X|w)$ from the phoneme sequence in the vocabulary calculated by equation 2.3 [7].

$$w* = argmax_w(p(X|w)p(w)) \tag{2.2}$$

$$p(X|w) = argmax_s(\prod_j(p(x|s_j)p(s_j))) \tag{2.3}$$

It is, however, not possible to calculate all possible probabilities as the possible states sequences grows larger. To solve this the Viterbi search algorithm can be used [13] as an optimal recursive solution, estimating the most likely sequence of states. This solution outputs a list of possible hypotheses sorted by total score, n-best list, and not just a single sequence of words [7] [13].

SR systems usually, in the **post-processing** step, attempts to re-score this list, e.g. by using a higher-order language model and/or pronouncing models [7].

2.4 Hidden Markov models

Hidden Markov model (HMM) is defined as *"a doubly stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols"* [14]. Stochastic process means, in probabilistic theory, a random process or a collection of random variables.

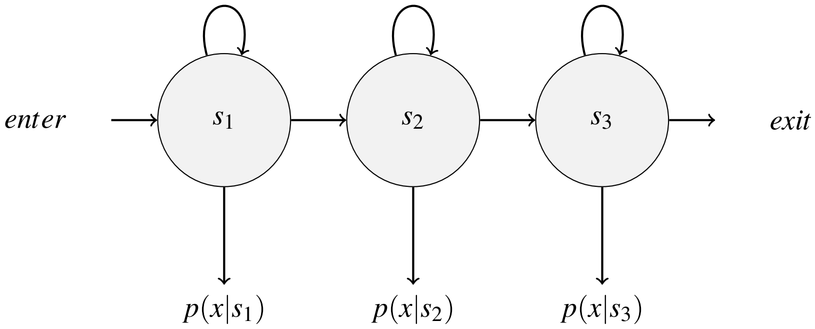

Figure 1: A three-state hidden Markov model

Figure 1 shows an illustration of a three-state HMM where each state s_i has a probability density $p(x|s_i)$ that states the probability density for the acoustic observation x for the state s_i. The three states s_1, s_2 and s_3 form the word S. HMMs is trained on speech data and if that data comes from a sufficient number of speakers, the model can be considered to be speaker-independent [7]. HMMs can be used in acoustic modeling for spectral and/or temporal analysis [15].

2.5 Gaussian mixture models

The Gaussian mixture model (GMM) is commonly used for determining how well each HMM state fits a frame of the acoustic input, i.e. the probability, and with enough components, they can model probability distributions to any level of accuracy. The accuracy of a GMM-HMM system can be improved further with fine-tuning after it has been trained [16]. GMMs model the probability density function in similarity to neural networks and is used in recognizing patterns [17], e.g. sound signals.

2.6 Neural networks

Artificial neural networks (ANN) are, as the name implies, inspired by the sophisticated functionality of the human brain where neurons process information in parallel. ANN consists of an layer of input nodes, then one hidden layer of nodes and finally an layer of output nodes [18], illustrated in Figure 2. Deep neural networks (DNN) adds more hidden layers to that. Most SR systems use HMMs to deal with temporal variety and GMMs to determine how well each HMM state fits a frame of the acoustic input, i.e. the probability, but DNNs has recently been proven to outperform GMMs on a variety of benchmarks [16] and are now used in some way by many major commercial SR systems, e.g. Xbox, Skype Translator, Google Now, Apple Siri, etc.

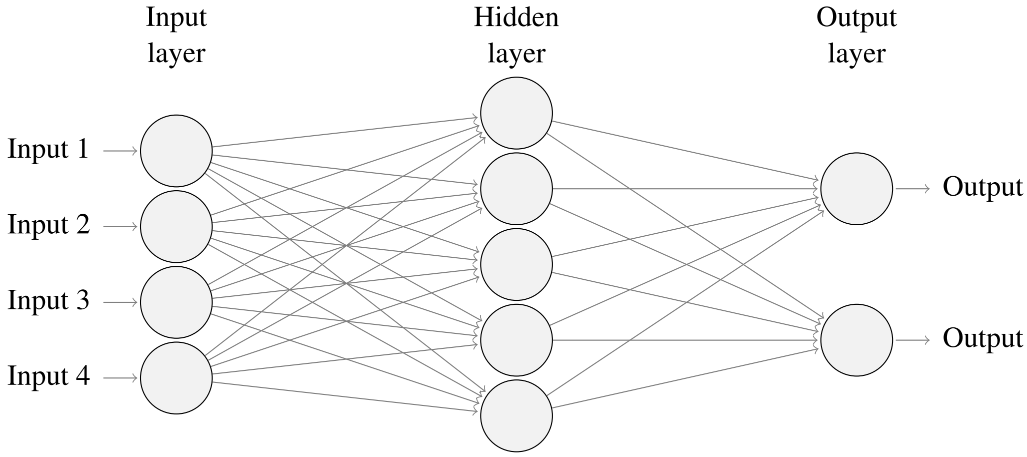

Figure 2: Illustration of a possible neural network

Every connection between two nodes in Figure 2 has a weight attached to it which is tuned while training to get the desired outcome [18].

2.7 Google Speech

Google's first attempt at speech recognition came in 2007, and was called GOOG-441 [19]. It was a speech-enabled business finder which took in user spoken city and state, followed by a business name or business category. The system would then try to recognize the query, check the entry against the web-based business search engine Google Maps which would return a list of business. Based on how close the match was to the query, one to eight results were read back to the user, using text-to-speech (TTS). The user then had the option to get connected to the service or request an SMS with the business information.

Google's current speech recognition system is speaker-independent and is built on deep neural networks together with hidden Markov models (DNN-HMM) [20] [21]. The strength of Google Speech lies in general purpose use, e.g. when making search requests on the world wide web. Google uses cloud-computing for speech recognition tasks.

Pocketsphinx on the other hand is speaker-dependent and is built on gaussian mixture models together with hidden Markov models (GMM-HMM) [22]. Pocketsphinx performs all computations locally.

3 Methods of measuring

Below is described how the evaluation of Google Speech was done, which data that was chosen to cover the different aspects of SR accuracy and information about the WER calculations the evaluation were based upon.

3.1 Data collection

To be able to test the two SR systems with different speakers, different lengths and with or without added background noise, a list of 100 sentences was made. One could argue that difference in pitch should be among these variables, but that would cause an combinatorial explosion not suited for this paper. The sentences were chosen for being common English sentences, with the addition of parts from The Rainbow Passage [23]. The Rainbow Passage is used by speech pathologists as it contains every sound in the English language [24]. The complete list of sentences used can be seen in Appendix A.

To get a reasonable diverse set of data, the choice was made to have 50 short (less than 7 words) and 50 long (more or equal to 7 words) sentences. These 100 sentences were then recorded by two Swedish speakers in the same conditions; in the same room, using the same uni-directional mic with the same mic placement and the same recording hardware/software.

Background noise, a recording of a crowd talking in English, were then artificially added to these files, creating copies. The choice to use the sound of people talking was made due to the most probable application scenario for this voice-controlled robot is either in a relatively silent location or around people, using the same reasoning as in another evaluation [25]. This resulted in a total of 400 audio files (100 sentences times two speakers times two for added noise files).

3.2 Implementation and calculation

A C++ program was built to do the actual measurements of time and accuracy, with its functions being sending the wave file to the SR system, extracting the resulting hypothesis whilst measuring the time to do so. The program then proceeded to calculate the WER value by using an implementation of the Levenshtein distance [26], calculating the minimum number of substitutions, insertions or deletions that has to be made to get the reference sentence. The testing sessions was also done under similar conditions using the same computer, as the CPU can affect computational time, and on the same internet connection. Specifications of the setup used for recording and testing is shown in Appendix B.

The reference sentence, hypothesis, time (in milliseconds), number of words, number of differences between reference and hypothesis and the calculated word error rate were then all written to text files.

These output files were then processed by a couple of bash scripts, calculating total WER and differences in translation time.

3.3 WER

Perplexity has been known for a very long time as a quality measurement for language models. Word error rate has been shown to have a strong correlation to perplexity through a power law [27]. WER provides no information about the specifics in the translation fault, but it is a valuable tool for measurements between different SR systems.

WER is derived from the Levenshtein distance, also called Edit-Distance, which calculates the least number of edit operations necessary to modify one string to get another [26].

An example of calculating WER value on a hypothesis sentence to obtain the reference sentence is shown below in Table 1 where the resulting WER would be (1+1+1)/6 = 0.5 = 50%.

Table 1 ”S” substitution, ”I” insertion, ”D” deletion, ”=” match

Reference	this	is	a	paper	about	ASR	
	=	D	S	=	=	=	I
Hypothesis	this		of	paper	about	ASR	system

4 Results

This chapter summarizes the results outputted by the program introduced in Chapter 3, showing the results in WER percentages and translation times.

4.1 WER comparison

Tables 2, 3, 4 and 5 shows comparisons between Google Speech and Pocketsphinx in regards to WER and the mean translation time in milliseconds. These includes recorded files from both speakers.

Table 2 Short sentences (six or less words per sentence)

System	Tot. words	Tot. faults	WER	Time (ms)
Google Speech	462	26	5,62%	2889,45
Pocketsphinx	462	158	34,19%	1381,19

Table 3 Long sentences (seven or more words per sentence)

System	Tot. words	Tot. faults	WER	Time (ms)
Google Speech	1024	118	11,52%	4215,75
Pocketsphinx	1024	428	41,79%	2387,18

Table 4 Short sentences with artificially added noise

System	Tot. words	Tot. faults	WER	Time (ms)
Google Speech	462	42	9,09%	2962,44
Pocketsphinx	462	255	55,19%	2653,33

Table 5 Long sentences with artificially added noise

System	Tot. words	Tot. faults	WER	Time (ms)
Google Speech	1024	130	12,69%	4495,19
Pocketsphinx	1024	539	52,63%	4177,62

Table 6 shows comparisons between speakers in respect to the two systems. This includes all recorded audio files per speaker.

Table 6 Speaker 1 vs. speaker 2

System	Speaker	Tot. words	Tot. faults	WER	Time (ms)
Google Speech	Speaker 1	1486	177	11,91%	3731,840
Google Speech	Speaker 2	1486	139	9,35%	3549,575
Pocketsphinx	Speaker 1	1486	714	48,04%	2886,865
Pocketsphinx	Speaker 2	1486	666	44,81%	2412,795

5 Controlling a robot with SR

AmigoBot [5] was used for testing the use of SR for controlling a robot in real-time. It is a differential-drive research robot with eight built-in sonars for object detection and is fitted with a wireless network interface card for communication.

The objective was to see if SR was effective in controlling a robots movements and which attributes was the most important when dealing with real-world moving robots.

The code was co-written with Rickard Hjulström and is written in Python, C++ and Bash.

5.1 The Robot Operating System (ROS)

The Robot Operating System (ROS) is used to communicate between nodes using messages through topics in a peer-to-peer (P2P) network [28]. A brief introduction to these concepts is given below followed by a figure of the communication (see Figure 3).

- Nodes
  Nodes are processes that perform computation. ROS is highly modular and one robot control system normally contains many different nodes, e.g. one node performs voice recognition and another controls the navigation.
- Master
  Nodes uses the ROS Master to find each other, exchanging messages or invoke services. When a node has found another node they connect directly to each other, making the Master work like a DNS server.
- Parameter Server
  A part of the Master and allows for data to be stored by key in a central location.
- Messages
  Communication between nodes are done through messages. These can be data structures of standard primitive types (integer, float, char, etc) and arrays of such types.
- Topics
  Messages are handled by topics. Nodes can publish (send) messages to them and nodes can subscribe to them, in order to receive messages. Topics can have multiple publishers as well as subscribers and a node can publish and/or subscribe to multiple topics.
- Services
  An alternative for the publish/subscriber system that instead uses request/reply interactions which is more appropriate for distributed systems.

- Bags
  A way to save data during run-time, such as sonar readings, and then be able to playing them back to the system. Very useful for testing and debugging.

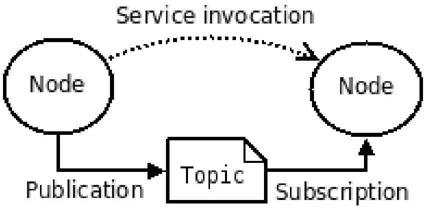

Figure 3: ROS communication [29]

5.2 SR in robotics

SR in robotics has been used in medicine as an aid in performing surgeries [3][2] and for educational purposes for the visually impaired [4] and it is clear that the use of SR in aiding humans is very useful. A study on using vowel sounds for controlling a robotic arm [30] to improve the abilities, independence and overall quality of life for individuals with motor impairments has shown that, with further research, a system could be created to aid these individuals using SR.

When controlling a moving robot in any way, the safety of humans is of course important. In this case, that is not a real issue as the robot is very small, but the same logic applies here because it also means keeping the robot safe and not perhaps tumbling down stairs. This can possibly be avoided using large number of sensors to allow the robot to make an internal model of the world, but as the real world cannot be interpreted in full due to the frame problem [31], we still need a way to communicate to the robot that we want it to stop when it's about to do something we know will be a mistake. We also expect that this would happen as soon as we spoke the word ”stop”, in the same way we expect any human we're communicating with to hear what we say the moment we say it. This makes the response time very important, not only for safety reasons but also for allowing us to naturally speak to technology.

5.3 Implementation and testing

A couple of algorithms were implemented to see where SR had the most use; moving forward, backward, left and right, following left or right wall, go home, save new home and stop. The importance of short response times became obvious at an early stage as the sonars were not yet implemented and so the robot were not autonomous. Although Google Speech measured more accurate than Pocketsphinx in an out-of-the-box state without any training and two speakers, the response time from Pocketsphinx were up to two times faster (see Chapter 4 for data) and therefor a decision was made to use Pocketsphinx for the voice-controlled robot. Pocketsphinx also allows for a custom vocabulary which in turn increases the accuracy, as mentioned in Section 2.2, for it only needs to contain implemented robot commands. The limitations coming from the need for internet access in order to use Google

Speech also contributed to the choice of Pocketsphinx.

The implemented system uses a simple keyword spotting system [32] to filter out the users normal utterances from the important commands and a flowchart of the commands is shown in Figure 4. Pocketsphinx's vocabulary now only contains these sentences below making it very small which should increase the accuracy.

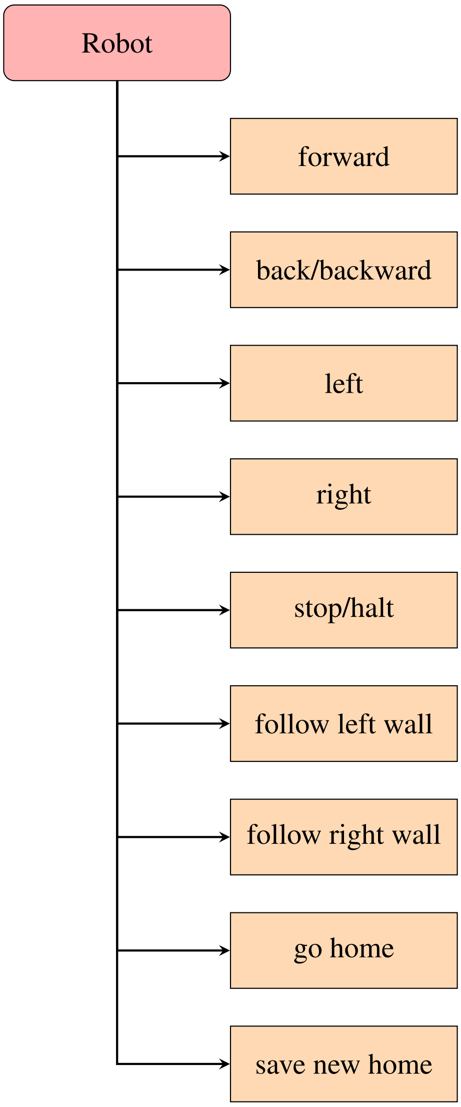

Figure 4: Commands flowchart

5.4 Robot commands explained

All commands must, due to the keyword spotting system, be preceded by the phrase ”robot” and can then be followed by a number of commands:

- forward
  sets a constant positive linear speed.

- back/backward
  sets a constant negative linear speed.

- left
  increases the angular velocity.

- right
  decreases the angular velocity.

- stop or halt
  sets linear speed and angular velocity to zero.

- follow left/right wall
  keeps the robot within a set distance from the left or right wall depending on the command while moving forward and following any turns.

- go home
  navigate to the last known home position, starting at point (0,0).

- save new home
  sets a new home position at the current position.

5.5 Data collection and results

New measurements with the robot commands vocabulary were made by recording all available commands in the same fashion as in Section 3.1 with the same two speakers. This resulted in another 44 audio files (11 commands, 2 speakers and added noise) and the results are published in Table 7, including both systems using files recorded by both speakers and with added noise in Table 8. The list of sentences used can be seen in Appendix A.1.

Table 7 Robot commands

System	Tot. words	Tot. faults	WER	Time (ms)
Google Speech	58	23	39,65%	3202,95
Pocketsphinx	58	3	5,17%	145,00

Table 8 Robot commands with added noise

System	Tot. words	Tot. faults	WER	Time (ms)
Google Speech	58	32	55,17%	3398,00
Pocketsphinx	58	5	8,62%	166,45

The time between getting the finished translated sentence and it being sent to the robot were measured programmatically to being between 1-10 ms, implying that translation speed is the dominant factor for latency.

6 Conclusions

The questions asked at the beginning of this thesis was:

- How does Google Speech compare to another speech recognition system regarding word error rate and translation speed?
- Which system is best suited for controlling a robot using voice commands?

Based on the results in Chapter 4, more precisely the overall translation times, Pocketsphinx is the better choice for implementations where low latency is of high priority.

Tables 2 through 5 in Chapter 4 show that Google Speech displayed 28,5%, 30,27%, 46,1% and 39,94% lower WER scores than Pocketsphinx.

Tables 2 and 4 show that short audio files containing background noise increased WER with 21% with Pocketsphinx and only 3,47% with Google Speech. With long audio files (Tables 3 and 5) the difference were not as noticeable, resulting in an increase of 10,84% with Pocketsphinx and 1,17% with Google Speech. These results indicate that Google Speech can filter background noise more effectively than Pocketsphinx.

The difference between speakers in Table 6 is to be expected due to difference in pronunciation and speaker styles, e.g. which words the speaker chooses to emphasize.

Pocketsphinx displayed 1,5-1,8 seconds faster translation speeds than Google Speech when it comes to clean short and long sentences, but it is also shown that given background noise Pocketsphinx's translation speeds almost doubled while Google Speech's only increased by approximately 300 ms, supporting the claim that Google Speech handles noise more effectively.

As for the robot commands, Table 7 in Section 5.5 shows that with a limited vocabulary Pocketsphinx takes point on both WER scores and, as expected, translation speeds. Pocketsphinx shows a staggering 20 times faster translation speed than Google Speech as well as 34,48% lower WER.

With the robot commands including background noise shown in Table 8, as in the previous results, an increase in both WER and translation times compared to the clean audio files can be seen.

If the use is of a general purpose nature, such as dictating sentences of all types, Google Speech with its low error rates close to perfection is a good choice. Especially when taking in consideration that none of the speakers in this work speaks native American English.

To answer the question ”which system is best suited for controlling a robot using voice commands?”, since response time is of high priority in voice-controlled robotics, Google Speech through Speech API leaves something to be desired. The results using the modified vocabulary used for our voice-controlled robot implementation is very clear on that

Pocketsphinx is the SR system, of these two, best suited for the task.

7 Discussion and Future work

When considering the results of the comparisons, one should keep in mind that all the first sets of measurements was made on systems in out-of-the-box states. It is stated in Section 2.2 that a speaker-dependent system, such as Pocketsphinx, tend to have lower WER scores than a speaker-independent system, such as Google Speech, when trained properly. This means that with, perhaps extensive, training the accuracy of Pocketsphinx would be higher than the results shown.

It is also stated that the vocabulary size plays a big role in the outcome. The possibility to tailor the vocabulary can therefore increase the accuracy, lowering the overall WER even more since the probability of correct pattern matching increases. This is very noticeable when comparing the results from the first set of tests with second ones.

Google Speech had approximately four to six times lower WER scores in the first tests and handled background noises more effectively than Pocketsphinx, making it a very accurate system that doesn't require any training. The Speech API does, however, require an internet connection and a developer authorization key with a maximum of 50 requests per day, making continuous SR a bad idea since that would mean sending requests constantly, filling up that quota quickly. In a commercial product, these constraints may not apply as the developer might pay Google to increase the quota or to get access to some other API.

In the results of the robot commands, it is important to know that with a very small vocabulary, any faults will drastically increase overall WER. Also the questionable grammar of the robot commands led to Google Speech rather making guesses like ”Robert Wright” instead of ”Robot right” as that would be more probable in a web search context, which is the area where Google Speech is normally used. This also comes from that we created the grammar for Pocketsphinx in this case, but in Google's case the grammar is not mutable making certain sentences very unlikely.

For future work, differences in pitch, talking speed and differences in background noise could be recorded and documented to get a more complete set of results. The acoustic model for Pocketsphinx could also be trained in different stages to observe differences in WER and if and when it surpasses Google Speech on general purpose sentences.

Bibliography

[1] Macmillan Publishers Limited, "speech recognition - definition and synonyms." `http://www.macmillandictionary.com/dictionary/british/speech-recognition`. Accessed May 20, 2015.

[2] H. Johanet, "[voice-controlled robot: a new surgical aide? thoughts of a user]," in Annales de chirurgie, vol. 52, pp. 918–921, 1997.

[3] S. Okada, Y. Tanaba, H. Yamauchi, and S. Sato, "Single-surgeon thoracoscopic surgery with a voice-controlled robot.," Lancet, vol. 351, no. 9111, pp. 1249–1249, 1998.

[4] S. A. E. Mohamed, A. S. Hassanin, and M. T. B. Othman, "Educational system for the holy quran and its sciences for blind and handicapped people based on google speech api," Journal of Software Engineering and Applications, vol. 2014, 2014.

[5] Adept MobileRobots, "Amigobot." `http://www.mobilerobots.com/ResearchRobots/AmigoBot.aspx`. Accessed May 19, 2015.

[6] J. Tebelskis, Speech recognition using neural networks. PhD thesis, Siemens AG, 1995.

[7] R. E. Gruhn, W. Minker, and S. Nakamura, Statistical pronunciation modeling for non-native speech processing. Springer Science & Business Media, 2011.

[8] Fifth Generation Computer Corporation, "Speaker independent connected speech recognition." `http://www.fifthgen.com/speaker-independent-connected-s-r.htm`. Accessed May 19, 2015.

[9] C. Chelba, D. Bikel, M. Shugrina, P. Nguyen, and S. Kumar, "Large scale language modeling in automatic speech recognition," arXiv preprint arXiv:1210.8440, 2012.

[10] A. J. Jerri, "The shannon sampling theorem—its various extensions and applications: A tutorial review," Proceedings of the IEEE, vol. 65, no. 11, pp. 1565–1596, 1977.

[11] U. Shrawankar and V. M. Thakare, "Techniques for feature extraction in speech recognition system: a comparative study," arXiv preprint arXiv:1305.1145, 2013.

[12] M. Westphal, "The use of cepstral means in conversational speech recognition.," in EUROSPEECH, 1997.

[13] G. D. Forney Jr, "The viterbi algorithm," Proceedings of the IEEE, vol. 61, no. 3, pp. 268–278, 1973.

[14] L. Rabiner and B.-H. Juang, "An introduction to hidden markov models," ASSP Magazine, IEEE, vol. 3, no. 1, pp. 4–16, 1986.

[15] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” Signal Processing Magazine, IEEE, vol. 29, no. 6, pp. 82–97, 2012.

[17] Y. Wu, “Gaussian mixture model,” Connexions, 2005.

[18] S.-C. Wang, “Artificial neural network,” in Interdisciplinary Computing in Java Programming, pp. 81–100, Springer, 2003.

[19] M. Bacchiani, F. Beaufays, J. Schalkwyk, M. Schuster, and B. Strope, “Deploying goog-411: Early lessons in data, measurement, and testing,” in Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pp. 5260–5263, IEEE, 2008.

[20] G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” Signal Processing Magazine, 2012.

[21] D. Steele, “Google talk neural networks for voice recognition.” `http://www.androidheadlines.com/2014/10/google-talk-neural-networks-voice-recognition.html`. Accessed June 5, 2015.

[22] C. Sphinx, “Basic concepts of speech.” `http://cmusphinx.sourceforge.net/wiki/tutorialconcepts`. Accessed June 5, 2015.

[23] G. Fairbanks, “The rainbow passage,” Voice and articulation drillbook, vol. 2, 1960.

[24] High Tech Center Training Unit, “Dragon naturallyspeaking - advanced.” `http://www.htctu.net/trainings/manuals/act/dragon_advance.pdf`. Accessed May 19, 2015.

[25] H.-G. Hirsch and D. Pearce, “The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” in ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW), 2000.

[26] The Levenshtein-Algorithm, “The levenshtein-algorithm.” `http://www.levenshtein.net/index.html`. Accessed May 19, 2015.

[27] D. Klakow and J. Peters, “Testing the correlation of word error rate and perplexity,” Speech Communication, vol. 38, no. 1, pp. 19–28, 2002.

[28] The Robot Operating System, “The Robot Operating System concepts.” `http://wiki.ros.org/ROS/Concepts`. Accessed May 19, 2015.

[29] The Robot Operating System, “Ros basic concepts.” `http://ros.org/images/wiki/ROS_basic_concepts.png`. Accessed May 19, 2015.

[30] B. House, J. Malkin, and J. Bilmes, “The voicebot: a voice controlled robot arm,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 183–192, ACM, 2009.

[31] P. J. Hayes, The Frame Problem and Related Problems on Artificial Intelligence. Stanford University, 1971.

[32] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, “A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems,” Industrial Electronics, IEEE Transactions on, vol. 52, no. 6, pp. 1478–1489, 2005.

A Recorded sentences

When the sunlight strikes raindrops in the atmosphere, they act like a prism and form a rainbow.
The rainbow is a division of white light into various beautiful colors.
These take the shape of a long round arch, with its path towering above.
There is, according to legend, a boiling pot of gold at one end.
People look, but no human ever finds it.
When a man looks for something beyond his reach, his friends say he is looking for the pot of gold at the end of the rainbow.
Throughout the centuries people have explained the rainbow in various ways.
Nations have accepted it as a miracle without physical explanation.
For certain groups it was taken that there would be no more general floods.
The Norsemen considered the rainbow as a bridge over which the Gods passed from earth to their dwelling in the sky.
Aristotle thought that the rainbow was caused by reflection of the sun's rays by the rain.
Please keep this secret.
This message doesn't make sense.
I never let anyone else feed my dog.
She was washing the dishes.
He accepted my present.
We appreciate your help.
I built this dog house.
Many people think that children spend too much time watching TV.
We traveled in South America.
I am staying at the hotel for the time being.
I can't afford to buy an expensive car.
He needs some new clothes.
I can't figure out why you don't like jazz.
I think it's time for me to move into a smaller home.
I'm begging you.
The wind blew her hat off.
I don't think it makes him a bad person just because he's decided he likes to eat horse meat.
You look nice with your hair short.
I think the time is right to introduce this product.
He broke into a house.
Don't worry about the past.
She had a basket full of apples.
It was Janet that won first prize.
She suggested that I should clean the bathroom.
She bought him a car.
I will wait here till he comes.

Make your airplane reservations early since flights fill up quickly around Christmas.
You've given me good advice.
Keep in touch.
I have to do a lot of things.
She went to see him the other day.
He adopted the orphan.
Thousands of dead fish have been found floating in the lake.
I just want to let you know that I think you're the most beautiful woman that I've ever seen.
Come and see me when you have time.
He is nice.
I have a lot to do today.
She asked him some questions.
There's almost no water in the bucket.
We'll have to camp out if we can't find a place to stay.
Don't believe what she says.
The kettle is boiling.
I don't think I have very good reception.
I started feeling this way last Monday.
I think it's unlikely.
Don't climb on this.
When I hear that song, I remember my younger days.
I didn't expect you to get here so soon.
She beat him to death.
It's about time to start.
I'll buy a new one.
I'll tell you a secret.
He can run faster than me.
This is the calm before the storm.
There is an urgent need for peace talks.
He explained in detail what he had seen.
Spring has come.
I am sure that he is an honest man.
He amused us with a funny story.
It's almost time to go to bed.
I'm free tonight.
I can't hide the fact from you.
For some reason, I'm wide awake.
I have to wash my clothes.
We ordered pink, but we received blue.
You must go.
Give me a coffee, please.
He had no choice but to run away.
You must study your whole life.
It was a terrible day.
I expect that Tom will pass the exam.
I had to postpone my appointment.
She kept her eyes closed.
I don't think that I deserved the punishment I got.

Something is wrong with this calculator.
You have a very nice car.
Education starts at home.
It’s up to you to decide.
She has a flower in her hand.
I know that she has been busy.
I’ll stay home.
I’ll see you next Wednesday.
That’s right.
See you tomorrow.
Plastic does not burn easily.
The books are expensive.
I lost my watch.
The prisoners were set free.
She killed him with a knife.

A.1 Recorded command sentences

Robot forward
Robot backward
Robot back
Robot left
Robot right
Robot follow left wall
Robot follow right wall
Robot go home
Robot save new home
Robot stop
Robot halt

B Setup specification

Setup used for recording and testing:

- CPU: Intel Core i5-3210M (2.5 GHz base clock, 3.1 GHz boost)
- RAM: 8GB (DDR3, 1333 MHz
- Operating system: Linux Mint 17.1 64-bit
- Internet: Fiber-optic connection (100 Mbit up/down)
- Microphone: SteelSeries Siberia v2