Speech-to-text: A Computer's Guide to the Human Language

Bryan Houston, Sam Otteskov, Savannah Vaultz

Supervisor: Ole T. Lassen

—

NIB 1. Semester Project, Roskilde University

December 22, 2014

Abstract

This report gives an overview of automatic speech recognition, with an investigation into dictation software for the purpose of understanding how it functions. In this report, we conduct an analysis of various speech input, as well as the explanation of the recognition process. This encompasses a more detailed look into what Hidden Markov Models are and how they are implemented within this process. Two speech recognition applications are presented—Windows Speech Recognition and Google Chrome's Online Dictation application—and tested for an understanding of the functionality and how the theory is applied in practical applications. These two speech recognition applications are given the task of transcribing isolated words, phrases, and then a continuous flow of speech, where two book excerpts are dictated to the applications. The report concludes with a discussion of the results and theory researched, in which factors affecting the accuracy and efficiency of SR technology are also discussed; it is inferred that these applications utilise Hidden Markov Models to recognise and transcribe speech, due to the properties of the models, because of their poor performance during the dictation of isolated words.

Contents

1 Introduction

The evolution of the information age is rapidly occurring, where the demand to improve society increases. As technology is progressing and improving, we as consumers are provided with a variety of options, making the technology convenient. The technology used varies from personal to work-related issues that can be solved with these improvements. The domain of technology spans over a wide spectrum, from personal computers to supercomputers. There are numerous devices available to help improve, as well as accommodate, our daily lives. The most common device in our era is the mobile telephone. Smartphone users are able to use a special type of application that allow them to interact with the device through just their voice. One notable such application is the virtual personal assistant, Siri, found on Apple's newer iPhones. These are called voice-user interfaces (VUI)[1]. What makes these applications unique is that they all have a form of *speech recognition*, or SR, within, that is, the identification by the computer of the actual words spoken given the recorded speech. Figure 1.1 illustrates an example of the functional steps within a VUI. The implementation of speech recognition within these applications is what enables people to vocally interact with them and the device.[2]

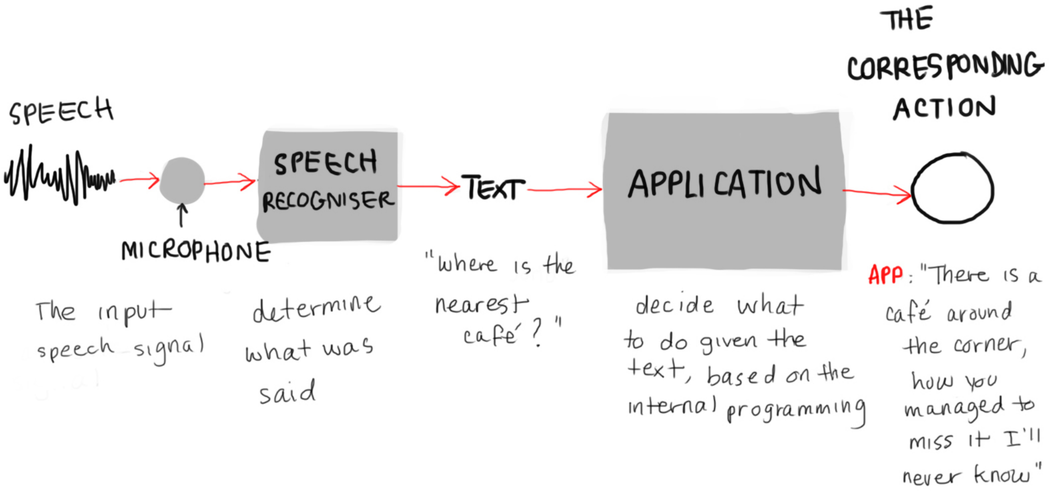

SPEECH
MICROPHONE
SPEECH RECOGNISER
TEXT
APPLICATION
THE CORRESPONDING ACTION
The input speech signal
determine what was said
"where is the nearest cafe ?"
decide what to do given the text, based on the internal programming
APP: "There is a café around the corner, how you managed to miss it I'll never know"

Figure 1.1: Overview of a voice-user interface, where the user asks for directions to the nearest café, the application going through a series of stages and finally giving a response, in which it replies to the user where the café is.

Such technology is also found in various other platforms besides the mobile, anywhere from on personal computers to the virtual helpdesk agents found in businesses' call centres. The convenience of speech recognition technology is applied to complete tasks in many different situations. These can be grouped into four main areas: voice control, dictation, dialog, and transcription [11]. Systems that utilise voice commands from a user to control a computer or some other application on it are grouped in the voice control task. iPhone's Siri is an example of this, where the user can ask it to find the nearest caf. Dictation involves the conversion of the words spoken by the user to text,

[1] for further reading, refer to [8]

[2] This will be explained in detail in chapter 4.

usually with some commands to allow for editing. Speech-based interactions between the user and computer fall under the category of dialog. The most difficult of the four tasks to achieve is transcription, where the system has to convert speech from multiple users who are not necessarily speaking directly to the system. Our focus is on SR that is utilised to complete the task of dictation.

These dictation applications can be used during medical examinations, or for letter writing when one would need their hands or eyes free for something else. These are but a couple of examples where a hands free solution can be applied. In this report, we narrow down the domain of speech recognition to that of *speech-to-text*, which is a form of speech recognition technology applied for dictation that translates speech into the written words it represents. It can be simplified as a function where what is going in is the speech, and what comes out are the recognised words.

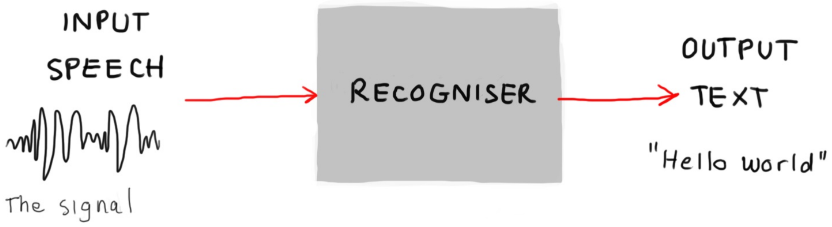


Figure 1.2: The recognition stage in a voice-user interface for dictation

Within these types of voice applications, there are a number of functional stages that are required for the system or program to work optimally. We will focus on one of the first stages, where the system must be able to interpret what the user says and print it out as readable text. As seen in figure 1.1, this particular stage is the recognition stage (the 'Speech Recogniser'). The three main parts involved here are seen in figure 1.2, and are:

· **Input speech** The speech sound signal produced by the speaker.
This is just a representation of the original words spoken in the form of sound waves or *acoustic signals*, generated when the user speaks into the application. This is what the application receives as the input.

· **'Recogniser'** The system that interprets and converts the speech.
This is the application itself where the *recognition-* or *transcription process* takes place, or the process of converting the input speech to its textual representation—much like when humans write down what someone has said.

· **Output text** The resulting transcription/recognised utterance or words.
The result of decoding the input speech by the recogniser is the *text*. This is the sequence of written words that the recogniser determines to be the best representation of the original words spoken.

When a user is dictating a letter, for instance, the application takes what they are saying and attempts (as best possible to the system's capability) to infer the words they are speaking and then transcribes them into the document as readable text. Two examples of such dictation applications are Windows Speech Recognition and Google Chrome's online Dictation application, and are tested as part of our project. It's very fascinating, that we can actually speak to a device and have it do something based on what we've said. This technology has been in development for the last few decades, since the 1930s, actually [19]. With the challenges that are involved in speech recognition, it is amazing that we can get machines to perform like this in the first place. Perhaps in the future, science fiction won't be fiction anymore—like is usually said for many different areas of science and technology.

We want to figure out how this is possible. How can it be made so that a computer can even 'understand' what is being said? We will be investigating part of a dictation VUI where the recognition process takes place. In figure 1.2, we want to know what is going on inside that grey box ('Recogniser'). Furthermore, we will be investigating this stage to understand how it works and what it can do, namely, attempting to answer the question: how does a computer convert a speech signal into the words that it represents?

Upon further consideration, is there a reason why this has been taking so long to develop? Apparently, creating an effective, efficient, and accurate speech recognition system is not that easy. In fact, it is quite difficult, and modern applications still show various discrepancies. Sometimes even the slightest wind can have an adverse effect on the accuracy of the transcription. It is difficult to make an effective speech recognition program because there are numerous factors that hinder its success (such as differing accents, speeds of speech, background noise, etc.), and thus make them troublesome to use. Perhaps speech recognition technology still has a long way to go yet.

1.1 Methodology

This report illustrates the functionality of a subset of speech recognition applications known as speech-to-text as they are applied for dictation tasks. The next chapter, chapter 2, first presents the problem formulation in section 2.1 and its subsections, upon which we conduct an analysis in order to determine the scope and direction of the project. We describe what exactly we will look into and our motivation and approach to this investigation. Important foundational concepts and definitions are explained so that we can reach an understanding of the necessary parts involved in finding a solution to the problem formulation. The two chapters that follow present more detailed descriptions of speech and speech recognition, respectively. Those two chapters together make up the main theory part of our research, illustrating our main focus.

In chapter 3, an analysis of the acoustic input to a speech-to-text system is demonstrated for the understanding of what exactly it is the computer has to process and transcribe. Visualisations of various speech signals are presented to help illustrate how differences in speech can affect the input signal to an SR system. After learning of the properties of these speech waveforms and how they represent different speech sounds, we investigate the technology itself in chapter 4, starting with how it actually processes this information. We learn about the architecture of a typical speech recognition system, and the different ways in which it classifies the speech data into a string of words.

Chapter 5 identifies the two specific applications of speech recognition technology for dictation, illustrating their design and uses: Windows Speech Recognition and Google Chrome's online Dictation application. A handful of tests are carried out in chapter 6 to see how speech-to-text operates in real-world applications. The tests are conducted on the two applications chosen for investigation.

The final chapters, 7 and 8, provide discussions of the research and problem formulation, and closes with a conclusion to the project. A glossary can be found following the Conclusion on page 51, in which we have placed important terms we use regarding our investigation, all of which have already been introduced and defined in the report. The final pages of the report consists of the Appendices, labelled A through E in which we list: all of the texts used during the tests[3], the collection of visualisations used for speech analysis[4], and the results of the dictation tasks for Windows Speech Recognition and Google Chrome's online Dictation application respectively[5].

[3] Found in Appendix B on page 56

[4] Found in Appendix C on page 57

[5] Found in Appendix D on page 60, and Appendix E on page 64

2 Preliminaries

In order to carry out our investigation, we must know exactly what it is we are trying to accomplish, and what we will need to do to accomplish that. We have formulated a research question in the next section, 'Problem Formulation', which we will try to find an answer to throughout this report. After stating our problem formulation, we break it down into smaller questions in order to more clearly define the direction of the project. Using those questions as a guide, we attempt to differentiate between what we are going to look into, and what we believe we must leave out—we ask ourselves: what is most conducive and relevant to the successful completion of our project?

2.1 Problem Formulation

> *What is done to make it possible for computers to* recognise speech*,*
> *that is, to convert speech into its textual representation?*

2.2 Problem Analysis

We have formulated a few questions that should help us in our investigation and hopefully allow us to fulfil our goal of understanding the inner workings of speech-to-text software. The aim of our project is to find out what the speech recognition application is doing to convert a given speech signal into the corresponding words it represents as text.

Below are the sub-problems to aid our investigation:

- What is the input—the acoustic signal—and how is it produced?
- How do computers analyse the acoustic signal?
- What is the structure/architecture of the recognition software?
- How does each part in a typical system function?

It is also necessary to evaluate how well this solution has solved the problem. In an effort to do so, we will also discuss some of the factors that affect the accuracy of such technology. We hope that we can achieve this through the questions we have posed below to lead the discussion on how effective applications of speech-to-text technology are today.

The discussion questions are thus:

- What limitations do systems face?
- Why are these systems, and others like them, not applied more?

2.2.1 Limitations and Fine-tuning

In this section we conduct an analysis of the problem formulation and sub-problems to clear up ambiguities and define the scope of the investigation. This is done for the purpose of determining what we can and cannot realistically do, what is within the scope of this project, and what is relevant.

Consider the problem, essentially: how does speech-to-text technology let the computer recognise the sound waves as speech and represent that information as text in a consistent way? To guide our analysis, we use figures 2.1 and 2.2; the first showing an overview of the process in a voice-user interface and the second, an overview of what goes on in the 'recogniser' shown before in figure 1.2.

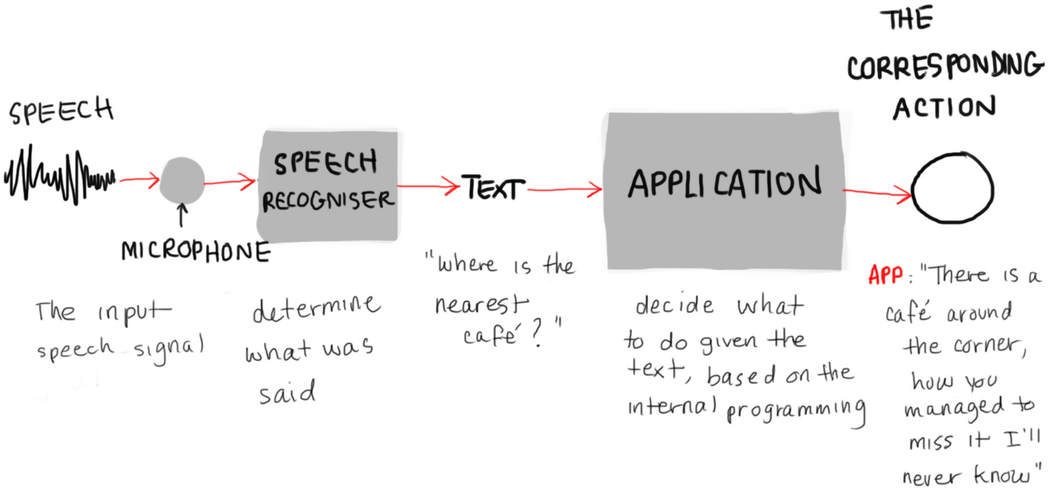

SPEECH
MICROPHONE
SPEECH RECOGNISER
TEXT
APPLICATION
THE CORRESPONDING ACTION
The input speech signal
determine what was said
"where is the nearest cafe' ?"
decide what to do given the text, based on the internal programming
APP: "There is a café around the corner, how you managed to miss it I'll never know"

Figure 2.1: Overview of a voice-user interface, where the user asks for directions to the nearest cafe, the application going through a series of stages and finally giving a response, in which it replies to the user where the cafe is.

Speech recognition (SR), also known as *automatic speech recognition* (ASR), is what allows a computer to identify and classify the words that a person utters [18]. It is the process by which a computer identifies the words within given audio data without the help of humans in a consistent way [11]. This was created to allow machines or devices to recognise and understand speech. And thusly, to allow for the interaction between such devices and humans with just the voice, in an attempt to bridge part of the gap between the user and the machine.

There are a number of steps in the *recognition process*, that is, the process by which the recogniser infers the original words spoken from the given speech signal Starting from one end of the process [14, 9], we have the acoustic signals (the speech) that are the information going in—the input—from the microphone or receiver. Next is the 'speech recogniser', the recogniser is part of the speech-to-text system that transforms the information and turns it into data that can be analysed. After this step is where the system attempts to recognise or infer what has been said by comparing or matching the data to words from an internal dictionary, based on a model or technique,[1] and finds the best match.

After (hopefully) successful transcription of the spoken words into text, the voice in-

[1] Chapter 4 provides an explanation of such a model, Hidden Markov Model, in section 4.2.1

terface program then takes this text and does some action based on what it says. In a dictation application, this would be to simply print out the resulting text into the selected text field (the document). We discuss the aspects of the SR system that we will study in the next paragraphs headed by: ‘The Input: Speech’, ‘Processing’, and ‘Recognition and Transcription’.

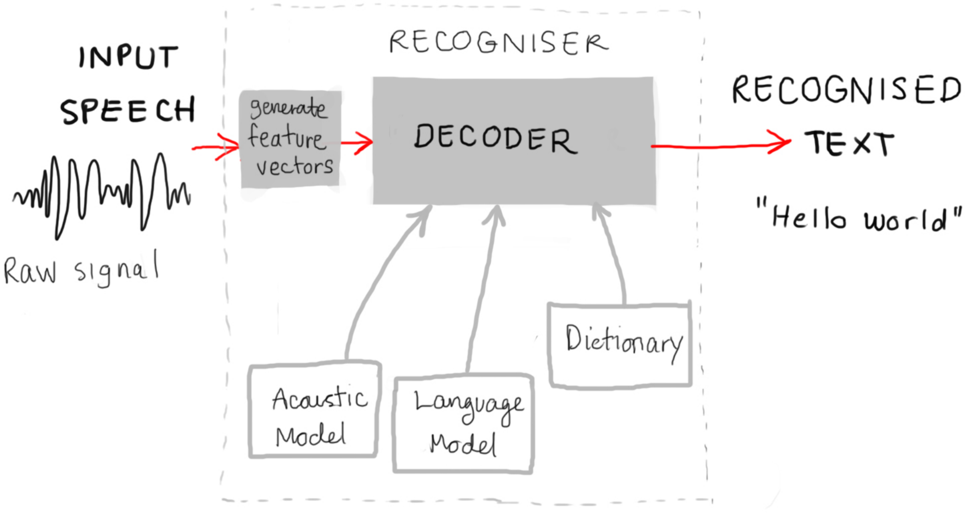


Figure 2.2: Components of a Speech Recogniser (adapted from [19])

The Input: Speech Understanding the mechanics behind speech itself will help us during our investigation. It may even be helpful to know a bit about how it is created and classified, as well as to know its components. This is where we take an approach to a branch of linguistics that studies exactly these aspects.

> *Phonetics*, as defined by Ball and Rahilly, is ‘the scientific study of speech’ [4]. It involves, to a degree, knowing how speech is produced, what particular characteristics the sounds have, and how others hear and perceive them. More importantly, phonetics involves the study of these particular speech characteristics, what they are made up of, how we produce them, as well as the acoustic characteristics (what characteristics the sound wave has when travelling in the air). Phoneticians are also interested in how speech is heard and decoded by the one hearing it, though this part we are not interested in.

We do not want to focus on the psychological, neurological, or chemical aspects of how speech is created. We want to focus on what makes up speech, and even, to a degree, how the different sounds are made and transmitted, so that we can know what it is that the computer has to deal with. What do the sound waves from speech look like? What is it that the computer is seeing when it receives this kind of information? In this case, it would be useful to be able to visualise this information with the Praat application so that we can have an idea of what the system is getting as input. Additionally, to answer our second sub-question, an analysis of some speech signals would also aid in our understanding the theory of what it is the computer would have to do during the recognition process.

Processing The recorded speech signal is what embodies the actual words spoken including all of the superfluous noises. When the system receives the speech signal from the microphone, it attempts to filter it and translate it into something that can be analysed and decoded. The signal data is broken up into small slices to be examined and decoded. We will look into how this is done to a degree, though we are more interested in how these pieces are matched with the words they are supposed to represent, and will not go into too much detail in this stage.

Recognition and Transcription Taking the data it has processed from the initial stage, it proceeds to convert this data into text that represents the original speech. We know that there are some statistical models used to determine what words or parts of words would best match the data the computer generated from the signal. A speech recognition system uses particular models to determine what words were spoken. One is the *acoustic model* which is what tries to match the words to specific parts of the speech signal. The other is the *language model* which is an algorithm that is used for determining plausible sentences. We would then need to know what these models are and how they operate within speech recognition systems (it might be best to consider the most-used or most popular models out there). We would also need to figure out how the system decides which words are the best match to the signal—which words would have most likely generated the signal given.

Models There are many different models that are used during the recognition process, and in some systems particular models are combined to carry out different parts of the decoding process. In this paper we try to understand a particular statistical model, called the Hidden Markov Model (HMM), and how it is applied to speech-to-text software and what role it plays in the recognition process[2]. We decided upon this model because it has shown to be the most popular of the models we have researched that is used for speech recognition.

Neural network models also have a role in speech recognition. This model is inspired by the connections of the neurons in the brain. The modelling systems have primarily been used in phoneme classification, isolated word recognition and speaker adaptation. Unlike HMM systems, the neural networks do not take into account statistical probabilities, they only try to match the phonemes by trying to fit a series of functions to the dataset. Neural Networks do not necessarily care about the probability of a particular phoneme following another like HMMs do. With the absence of these properties neural networks only work within a short spectrum, and we have decided not to go into depth with this particular model as it is not implemented nearly as much as HMMs are.

2.2.2 Systems

The following are the systems that we will make use of throughout our project.

Praat We will be using Praat throughout the project to help in our analysis of the acoustic signals we produce. Praat is a program that allows the user to analyse, synthesise, and manipulate speech, created by Paul Boersma and David Weenink (Institute of Phonetic Sciences of the University of Amsterdam)[3]. We will be using this program to help with our understanding and analysis of various acoustic signals.

The Speech Recognition Applications We have researched various SR software (ready-to-use software, since we have virtually no experience with using the various

[2] We will do this in section 4.2, where HMMs are discussed in section 4.2.1

[3] For more information on Praat, visit http://www.praat.org [6]

toolkits out there to create our own software) and compiled a short list of the ones we feel would be most appropriate to include in our project, ultimately choosing two:

- Windows Speech Recognition
- Apple's Speech Recognition for MacBook
- TalkTyper
- Google Chrome's online Dictation application
- Tazti
- Nuance's Dragon Dictation
- Samsung's mobile speech recognition software

We have decided to use Windows Speech Recognition and Google's browser-based Dictation[4] application, referred to as Dictation from now on, because they are good examples of both a simple and more complex SR, as well as for availability and financial reasons. Windows Speech Recognition, hereafter referred to as WSR, is readily available to us. We will illustrate these in detail in chapter 5.

2.3 Semester Requirements

As a first semester project at Roskilde University, there are some requirements that must be fulfilled. First, the project must conform to the semester theme, that is, the 'application of science in technology and society'. Secondly, the approach must be problem-oriented, as opposed to subject-oriented where the report would essentially be a presentation of information—a survey.

2.3.1 Semester theme

The theme requires us to identify, illustrate, and evaluate specific applications of science in society. Our chosen domain is that of speech recognition, roughly within computer science. By investigating the theory behind speech recognition, specifically how speech-to-text works in dictation applications, we have then been able to apply that in our evaluation of the two systems chosen. These systems, the specific applications of speech-to-text technology, are identified and illustrated in this report. They are then evaluated in the form of the various tests presented in chapter 6, the results of which are discussed in the 'Discussion' chapter. WSR and Dictation are just two among the numerous applications of speech recognition technology found in today's society, including Apple's Siri, and the newest as of now, Cortana, for Samsung's smartphones. All of these are generally designed to improve the effectiveness and efficiency in not only the dictation of documents, but also in planning, documenting, and other productivity tasks that people may have.

The sciences involved in the development of this technology includes mathematics (probability used to create the statistical models used when decoding speech data), computer science (the wealth of knowledge needed to develop these systems: algorithm design and analysis, signal processing, even machine learning, and more), as well as phonetics.

[4] https://dictation.io/

2.3.2 Problem-oriented

Additionally, our approach is problem-oriented, seeing as we have defined a clear question and the scope in which our report is designed to find the answer, or possible answers. Our project is driven by our problem formulation, and through that we have been able to lay out the groundwork for how we will move forward towards our goals.

2.4 Summary

Now that we know the scope of our project and what exactly we will look into, we can begin our investigation for a solution to our problem-formulation. In the next chapter, 'Phonetics and Speech', we will start off the investigation with an analysis of the acoustic input to a speech-to-text system so that we could understand what exactly it is the machine has to process and transcribe.

3 Phonetics and Speech

Keywords: *speech, non-speech, segments, syllables, acoustic signal, phone, phoneme, phonetics, acoustic phonetics, articulatory phonetics, voiced, voiceless, oscillogram, spectrogram, formants.*

We begin our investigation with an analysis of the acoustic input to a speech-to-text system. To understand what exactly it is the machine has to process and transcribe, it is imperative to learn of the properties of such signals and how they represent different aspects of speech. The following sections in this chapter include a discussion of the creation and transmission of speech.

3.1 An Understanding of Speech

While there are many different definitions for speech, it can be understood that speech is essentially a vocal representation of human communication—of language. Language is a symbol system that maps a word to an item or concept [4]. There are various ways in which such a word can be represented, whether that be as written words, sign-language, speech, or even Morse code. As you've been reading these words, you are gaining information through them via the written symbolisation of the language—in this case, English. If we were to speak these words to you, then the same information would just be understood through the speech symbol system instead.

Speech is one of the primary symbol systems of language. It is a verbal system of representation of the words and things languages are made up of. We will define it as the collection of sounds that are concatenated and grouped together in a particular way so as to form the vocalised representation of particular words or sentences in a language. There are very many, though a finite amount, of possible sounds that humans can produce. Portions of this collection of sounds that make up speech are found in the inventory of a particular language. [23] While there are many different possible sounds we can make, certain sounds, such as the raspberry-type noises, are not necessarily included in any known language. Thus, such sounds, as well as any other nonsensical vocalisations (meaningless to the known languages, at least), are to be considered as *non-speech* sounds hereafter.

3.1.1 Phones

We mentioned that speech is a collection of sounds found in human languages, these sounds are the various *segments* that speech can be broken up into. These collection of sounds are known as *phones*, or speech sounds. There are quite a lot of these[1], including sounds ranging from the 'ch' sound at the end of 'spee**ch**', or the 'th' sound in 'wea**th**er', to the various clicking sounds that can be made by the tongue and throat. The best-known system for transcribing these various phones into single symbols is the International Phonetic Alphabet (IPA)[2] [3], represented in brackets `[ ]` to denote that they are sound symbols independent of the language spelling.

[1] There are 40 phones in the English language [12]

[2] See https://www.langsci.ucl.ac.uk/ipa/index.html for more information about IPA, and Appendix A for a chart of the alphabet itself.

3.1.2 Phonemes

The phonetic symbols represent speech in the form of its individual phones, and can be grouped together to form classes of similar phones, represented as just single symbols called *phonemes*, represented in forward slashes `/ /`. In other words, phonemes can be considered 'generalisations' of phones, in that a phoneme is an abstraction of a set of phones. For example, the k-sounds in the words 'cat' and 'scratch' are pronounced differently and are transcribed as the phones [kʰ] and [k] respectively[3]. However, these sounds are of the same phoneme, transcribed as just a single /k/. Phonemes can be combined to form something called morphemes, as well as words. Morphemes are made up of combinations of phones or phonemes to form sounds such as the suffix 'ing' [ɪŋ] in 'running', made up of the 'i' [ɪ] and 'ng' [ŋ] phones, though they can be standalone as well, such as the morpheme `cat` or `push`[4] [23, 4]. These sounds are what the computer has to analyse and classify into words or segments that are representative of the original words spoken.

3.1.3 Phonetics

Phonetics itself, as we've defined in section **??**, is the study of the sounds that make up speech and how they are created/produced, what their characteristics are, how they are transferred, and how they are perceived and heard and then decoded. As mentioned before, we won't go into depth with how speech is perceived and decoded by the human, but rather the other parts of the process: how it is produced and transmitted. The sound of the phone naturally depends on the way in which it was made—how one uses their mouth and vocal organs to control the airflow and produce a speech sound. The study of these operations is known as *articulatory phonetics*—the study of the mechanics of speech.

3.2 The Mechanics of Speech

Articulatory phonetics refers to the study of the process of how speech is produced by humans, namely, how we manipulate our lungs, vocal cords, tongue, and mouth to create speech. We will focus more on the latter half (tongue and mouth control) as these are more interesting to our investigation. Refer to figure 3.1 for a visual of these and other parts of the speech system.

3.2.1 Initiation

Speech begins with the lungs, from where the air is pushed up through the trachea and larynx. Inside the larynx are the set of muscles called vocal folds, normally referred to as the vocal cords (sometimes the larynx is even called the 'voice box'). Sound is produced when air moves through the trachea and larynx, usually exiting through the mouth and/or nose; it could otherwise be stopped by an obstruction created by the mouth or other parts of the vocal system, as those depicted in figure 3.1. Continuous sounds are maintained with the actions of the diaphragm and the various muscles between the

[3] The superscript ʰ symbolises that the 'k' is aspirated.

[4] Where `push` as /pʊʃ/ is the sequence of phonemes /p/, /ʊ/, /ʃ/, and [pʰʊʃ] is the phonetic sequence of sounds [pʰ], [ʊ], [ʃ]

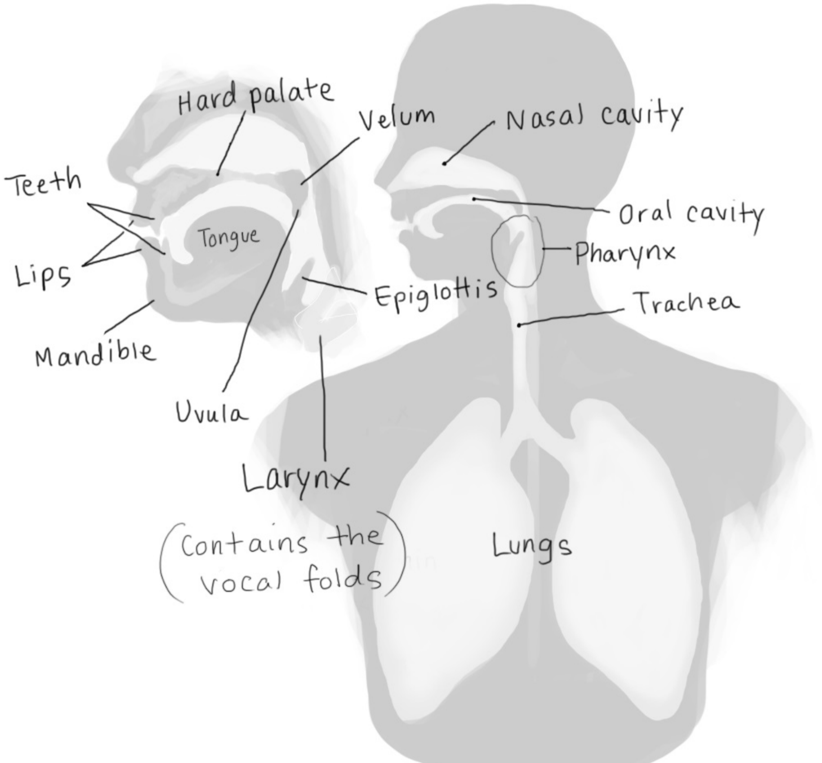


Figure 3.1: Parts of the vocal system (adapted from [23])

ribs, controlling the airflow in the lungs. If enough pressure is generated, then the vocal cords will vibrate, something like strumming a guitar. If one passes their thumb very slowly over the strings (a very slow strumming motion), no sound will be produced, as the pressure applied to make the strings vibrate is not enough. However, if they repeat this, gradually increasing the pressure and speed at which they run their thumb across the strings, at some point it will become enough to cause the strings to vibrate noticeably and produce sounds. Sound, however, doesn't necessarily have to be like this, like the humming sounds produced when we hum a melody. They can also be sounds like the ones made when whispering something. The difference between the two is simply that one is produced with vibrations in the vocal cords—voiced, whereas the other is not—unvoiced.

As the air is pushed out of the lungs by the diaphragm and muscles surrounding the ribs, it flows through two tubes that meet into one, called the trachea—otherwise known as the 'windpipe'. Further up lies a cartilage structure housing the vocal cords, called the larynx, above which are located the pharynx (the part of the throat located between the larynx and oral cavity as shown in figure 3.1) and the oral and nasal cavities. These three as a group are known as the vocal tract, and can be manipulated by the muscles to alter the way the air leaves the speaker. [23]

Inside the larynx are thin sheets of muscle that stretch out from its inner sides, forming the pair of vocal cords as seen in the drawings in figure 3.2 that can be stretched open or closed to alter the opening from the trachea. The vocal cords (represented as the dark, striated sections at the centres) are attached to two small pieces of cartilage at the back of the larynx (the bottom of each drawing in figure 3.2) that define the opening between the trachea and pharynx. The positions of these pieces are controlled by muscles attached to these and the surrounding cartilage.

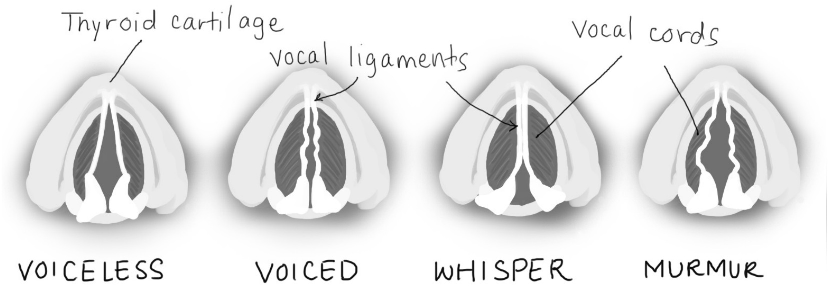


Figure 3.2: Four states of the opening between the vocal folds. Adapted from [23].

There are a few different ways in which the vocal cords can be, and the effect they have on the sound produced. If the cords are pulled open (as shown in the leftmost drawing in figure 6.2), air flows rather freely and smoothly through the opening, resulting in a breathy, or 'voiceless', sound with no vibration in the throat. Such a sound can be made when speaking words beginning with voiceless speech sounds such as the phones `[f, ʃ, h]` in the words '**f**ish', '**sh**eet', and '**h**ouse', respectively. [4, 23]

'Voiced' phonation is made through the vibration of the vocal cords. If the vocal folds are pulled lightly together and the air passing through causes them to vibrate, then the resulting sounds are voiced. Try to feel the difference between these two states when producing a continuous `[z]` sound, as opposed to an `[s]` sound, when saying the beginnings of the words 'zip' and 'sip'. Average vibration cycles for the cords in this state are 120 Hz for male speakers, and 220 Hz for female speakers [4], though these frequencies would of course change if they raised or lowered the pitch of their voices.

When 'whispering', the vocal folds are placed much closer together than in the voiceless state, though with an opening at the back of the larynx (refer to the 'whisper' drawing in figure 3.2). There is no vibration in the cords when air is pushed through.

In the last drawing, 'murmur', at the right in figure 3.2, the vocal cords are relaxed in a position that creates a wider opening than for a whisper, but narrower than for the voiceless state, vibrating when air is pushed through.

3.2.2 Articulation

Those were some of the different ways in which the air can be directed and controlled to create sounds, which can be further articulated through the manipulation of the airways in the vocal tract. An overview of the various airways in the vocal tract are presented in figure 3.3.

When we speak we activate the different parts of the vocal tract to form the sounds that we need to say words and sentences. For instance, pressing part of the back half of the tongue to the hard palate allows us to pronounce the 'g' in words like '**g**rass', '**g**reat' or '**g**alaxy', as well as the 'ng' in 'runni**ng**'. If you say these words, notice that the mouth is also in different positions when pronouncing them, and even the tongue is in a slightly different position as well for the last for a more nasal sound. Sounds like the 'ng' (written as the phone `[ŋ]`), such as `[n]` and `[m]` are also known as nasal sounds. Nasal phonation is pruduced with airflow through the nasal cavity, hence 'nasal' sounds, which are usually voiced. It's these little changes that determine what the sound will be

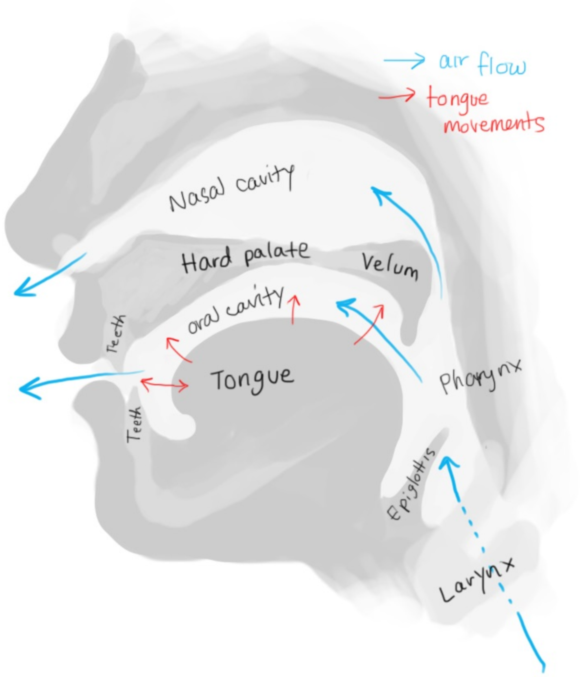


Figure 3.3: Airways in the vocal tract. The blue arrows signify airflow from one cavity to another. Red arrows about the tongue signify some possible movements that modify the airflow from the throat to the lips and out. For instance, by moving the velum to close off the path from the pharynx to the nasal cavity, air cannot flow through there and must travel through the oral cavity instead. The tip of the tongue could then be placed under the upper teeth, resulting in a either a voiced or unvoiced 'th' sound (/ð/ or /θ/ respectively).

like, and consequently, what the words will be.[5] Examples of these, along with others, with accompanying visualisations will be discussed in the following section.

3.3 The Acoustic Signals

This part of the speaking process is often referred to as the acoustic stage of phonetics, or the transmission of speech, the study of which is known as *acoustic phonetics*. The particular aspects studied in this stage involve the physical properties, such as amplitude, frequency, and duration of the sounds waves that represent the speech. So, what do the different speech sounds look like? We continue the discussion of articulatory phonetics combined with a look at visualisations of various speech input via the Praat program.

Consider the words: 'wreck a nice beach'. We recorded Bryan speaking these words and opened the audio file with the Praat program. Figure 3.4 is an oscillogram of the waveform, showing the amplitude of this acoustic signal over time.

We would like to separate this waveform into its individual speech sounds. First of

[5] Here is a clip of a real-time MRI of someone speaking that shows the movements of the different parts of the vocal system: http://commons.wikimedia.org/wiki/File:Real-time_MRI_-_Speaking_%28English%29.ogv [13].

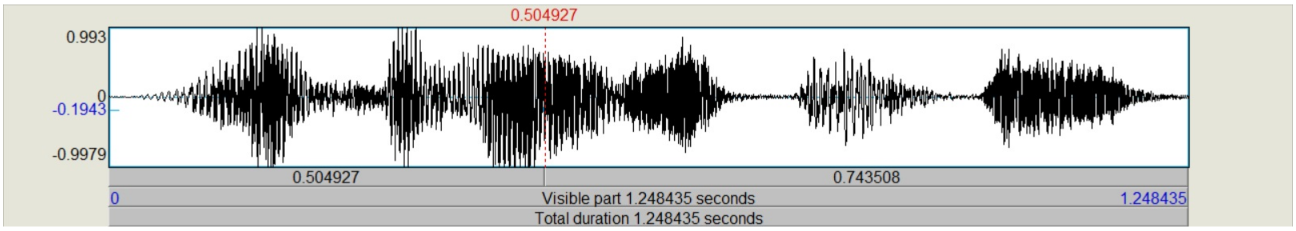
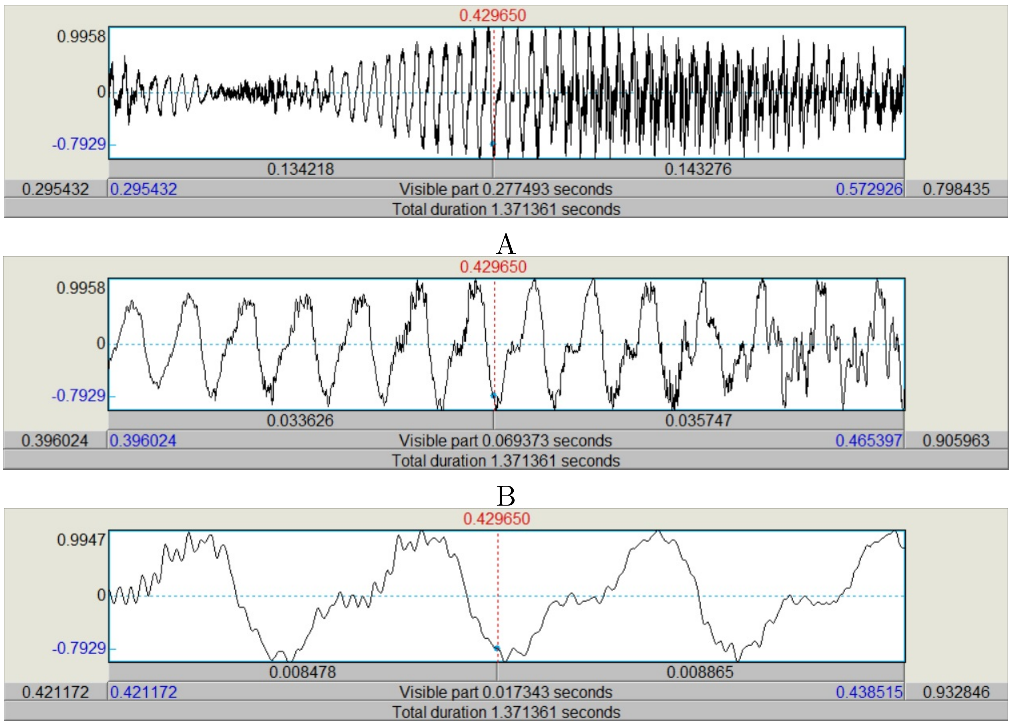

0.504927
0.993
0
-0.1943
-0.9979
0.504927
0.743508
0
Visible part 1.248435 seconds
1.248435
Total duration 1.248435 seconds

Figure 3.4: The waveform representation of the words ‘wreck a nice beach’ as spoken by Bryan.

all, it is very difficult to tell much apart from the waveform as it is. We would like to see the frequencies that are contained within the signal, because different vowels and consonants resonate at various frequencies. It turns out that the patterns of frequencies at which these sounds resonate are unique, but very similar for different speakers—that is, if they pronounce sounds the same way.

If you have a complicated but continuous waveform, it is actually a compilation or sum of simpler waves represented as sines and cosines. If we zoom in on part of the waveform, we can see many seemingly random jumps in the amplitude of the signal in part C of figure 3.5. These are just the result of many individual waves that, when added together, created this more complicated wave we’ve recorded.



A

B

C

Figure 3.5: Zooming in on part of the ‘wreck a nice beach’ waveform.

The Fourier transform is what is used to recover these component waves from the original signal by taking advantage of the properties of sine and cosine. A Fourier transformation, in this context, is essentially the process of breaking down an acoustic signal into its individual component waves at the frequencies they travel at. If we applied this to our waveform, what we would get is a spectrum of waves that oscillate at specific frequencies that make up the original signal in what is called a spectrogram. There can be a number

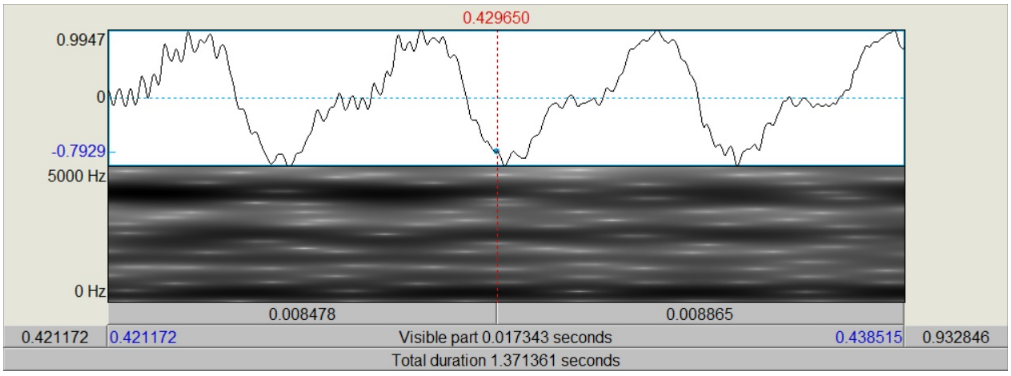


Figure 3.6: Spectrogram of part of the 'wreck a nice beach' waveform.

of waves that travel at a certain frequencies, and the intensity or density of such groups of waves are determined by the way the sounds were articulated. This intensity is represented in the spectrogram as dark patches, or 'formants', as shown in figure 3.6, where we've told Praat to show us the concentration of the component waves from the zoomed-in waveform in part C of figure 3.5. We apply this to the entire waveform representing 'wreck a nice beach', so that we can more clearly see the movements and patterns of the formants for the different sounds in the recording (figure 3.7).

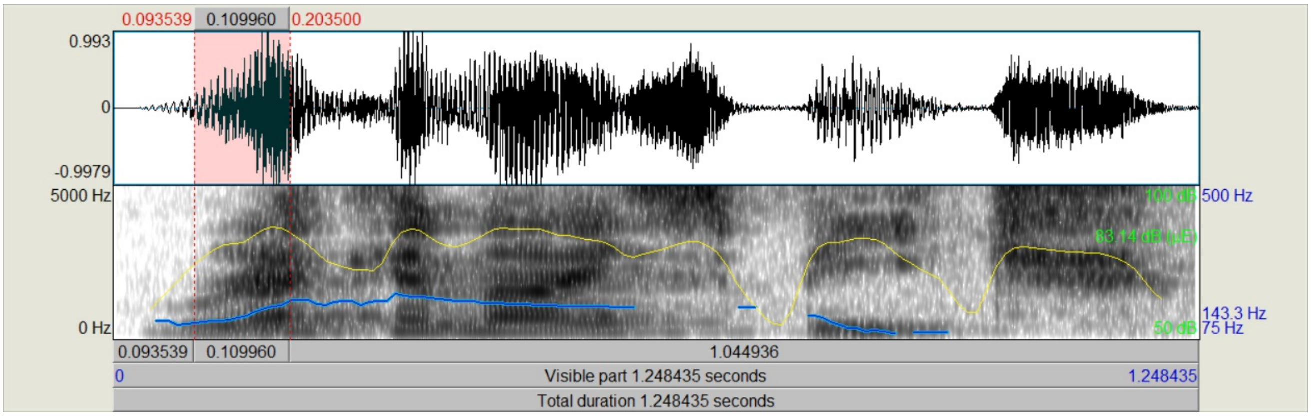


Figure 3.7: The waveform representing 'wreck a nice beach' after a Fourier transformation to see the concentration and intensity of acoustic energy at frequencies between 0 Hz and 5000 Hz shown in the spectrogram.

What is measured as frequency in acoustic phonetics, is perceived as pitch in auditory phonetics. This can be seen as the blue lines in the spectrogram in figure 3.7, where in the red highlighted area, the blue line goes up a bit to indicate a raise in pitch. The pitch can actually be calculated from the distance between the peaks in the wave, or the period, also represented by those vertical striations visible in the spectrogram. The intensity (in dB) is represented by the yellow lines in the spectrogram (the amplitude of the waveform in the oscillogram), and is perceived as loudness. The duration (time), is just the tempo at which people speak. Praat allows us to see each of these aspects for each recorded signal, which can be helpful if one wishes to know how the speaker alters the pitch or intensity to create various inflections in their speech, for instance.

We want to conduct an analysis on this waveform in order to understand and illustrate the process that a computer would essentially have to do in order to identify the phones within a phrase, given the recorded signal. Looking at the spectrogram in figure 3.7, there are some relatively clear differences in the patterns that these formants form, and it is possible to identify the boundaries that separate the various speech sounds. Each of the speech sounds generate a specific and unique pattern of formants. For instance, different vowels are characterised by different formant patterns—the articulation would cause the various sound waves to resonate at particular frequencies. Each speech sound would have a particular pattern of formants, areas of noise, as well as particular intensities of both. Let's mark off the boundaries that define the segments of individual speech sounds. This is done by simply identifying the spots in the spectrogram where the formant patterns change. For the 'wreck a nice beach' waveform, it can be broken up into ten different sections:

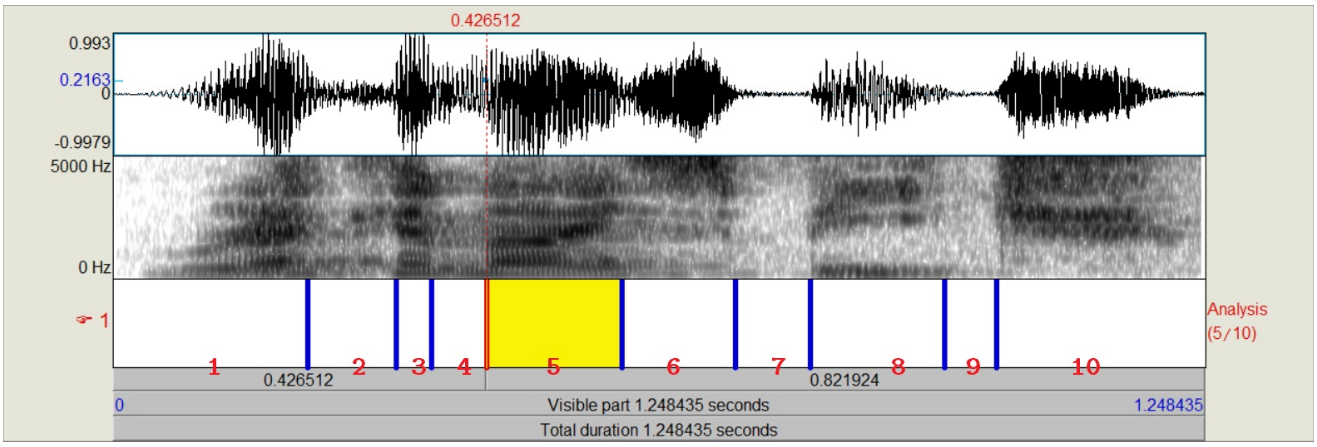


Figure 3.8: 'Wreck a nice beach' broken up into segments according to differences in formant patterns. These are numbered in red from 1 to 10.

The differences between voiced and unvoiced speech sounds can be seen in the absence of formants at the lower frequencies, the bottom of the spectrogram. One of the examples presented earlier was the difference between 'zip' and 'sip', where the /z/ and /s/ is voiced and unvoiced, respectfully. Sounds like this are known as fricatives, which are consonants produced with friction through nearly closed points in the vocal tract.[6] A clear example of this is shown in figure 3.9, where we've recorded these two 'z' and 's' sounds:

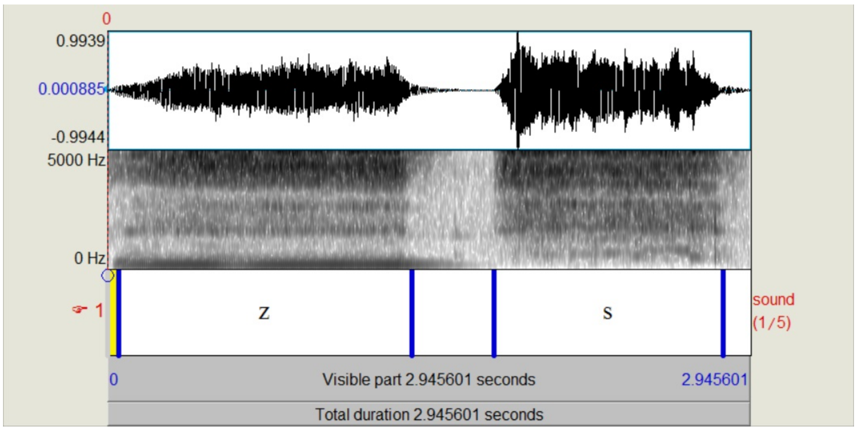

0
0.9939
0.000885
-0.9944
5000 Hz
0 Hz
1
z
s
sound
(1/5)
0
Visible part 2.945601 seconds
2.945601
Total duration 2.945601 seconds

Figure 3.9: The difference between voiced and unvoiced speech sounds, illustrated through a visualisation of the example of the 'z' and 's' sounds.

[6] This includes the English [f, v, s, z, ʃ, ʒ]. The [v], [z], and [ʒ] phones are the voiced versions of the unvoiced [s], [f], and [ʃ] phones. There is airflow through the mouth, and virtually none through the nasal cavity. [4, 23, Praat Manual]

There is great acoustic energy at the higher frequencies, and these two sounds are both characterised by a lot of noise in these upper frequencies (with the most energy at the highest frequency) [4, 23]. Notice there is actually a formant present at the bottom-most frequency for the 'z' utterance, and not for 's'. The same difference can be seen in our 'wreck a nice beach' recording, where in segments 2, 6, 7, 9, and 10 there is the absence of this formant at the bottom of the spectrogram. These segments would be the unvoiced sounds. Because there is a large concentration of noise in the upper frequencies in sections 6 and 10, these are most likely the 's'/'tch' sounds at the ends of 'ni**ce**' and 'bea**ch**' in 'wreck a nice beach'.

There are a couple of areas in the spectrogram that are more or less blank, where amplitude of the waveform is virtually zero (ignoring noise), like those in the 7th and 9th segments in figure 3.8. These are known as plosives, or stops in vocalisation where there is a complete obstruction in air flow. This usually occurs when pronouncing consonants in words[7]. For example, the /p/ in '**p**ush', or /k/ and /t/ in '**c**a**t**'. As mentioned earlier, the 'p' in 'push' and 'c' in 'cat' are aspirated, written as the phones [k^h] and [p^h]. We just represented this by placing the 'h's shown in figure 3.10 after the 'p' and 'k'.[8]

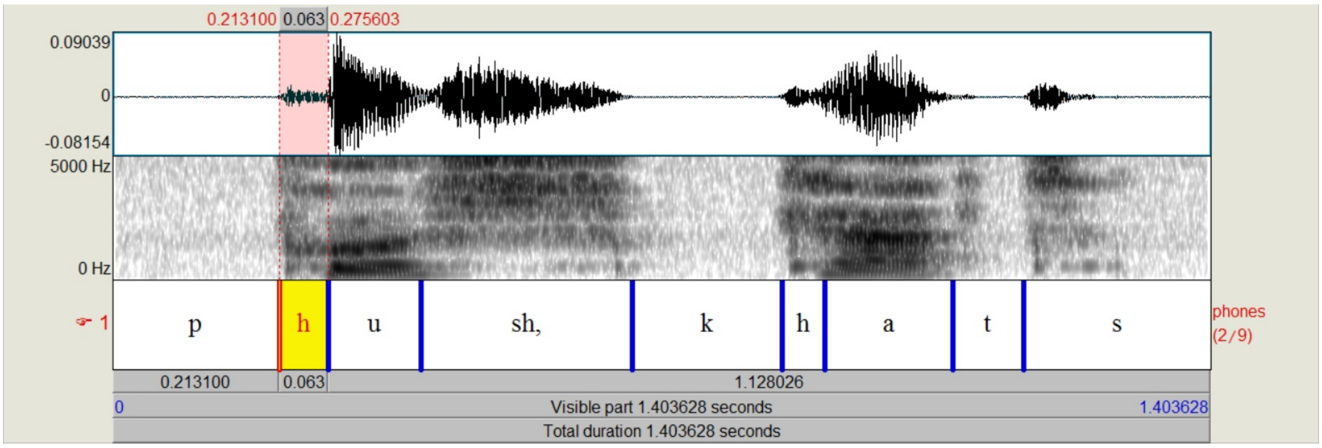


Figure 3.10: Annotated visualisation of: 'push, cat'.

Sometimes a sound is spread across two segments, or there is one section for two or more sounds. Listening to the part of the waveform in segments 9 and 10 (figure 3.8), we hear the 'ch' sound, which, in this case is spread across the two. In 9, the 'ch' begins with the /t/ plosive, and in 10, ends with the /ʃ/ fricative. Together these form the 'ch' in 'bea**ch**'. This is annotated accordingly in figure 3.11. Additionally, the 's' sound in the 6th segment is written under the spectrogram simply as the /s/ phoneme.

For the first segment, there is the 'wre' in '**wre**ck'—an example of a part containing more than one sound. This begins with the /r/, an example of an approximant (English examples: [w, j, l, r]), which are produced by a narrowing in the vocal tract, leaving just enough space for air to flow without much noticeable audibility. This and the rest of the 'wre' would be transcribed as /rɜ/. Notice that this is not preceded by a /w/, since this just naturally pronounced with the 'r' sound, rather than a 'w' sound as in '**w**ick', without the 'r'. This is followed by the /k/ sound in the second segment.

[7] This includes the phones [p, b, m, t, d, k, g] in English.

[8] The [k] and [t] sounds, are known as a velar voiceless plosive [k^h]—due to contact with the velum by the tongue, and a dental voiceless plosive [t]—due to the contact with the teeth by the tongue [4, 23, Praat Manual].

The following segment after that contains the sound for 'a', then followed by the 'n'. The 'a' as pronounced in this recording is transcribed as an upside-down 'v', /ʌ/. The /n/ formant pattern looks very similar to that of the /a/ in this case, since the only difference in articulation is that the tongue is making contact with the hard palate to redirect the airfow to the nasal cavity for the /n/ sound. We can see this difference in the intensity of the formants for these two segments, where the sound is blocked a bit by the tongue for the 'n' for a less intense sound than that of the 'a'. The 'n' in '**n**ice' is just transcribed in figure 3.11 as /n/.

In the fifth segment, the movement of the formants shows a change in articulation to produce the dipthong vowel 'i' in 'n**i**ce'. This here is written as the combined /aɪ/ in the figure. The 'b' in '**b**each' would just be transcribed as /b/. Lastly, we have the 8th segment, where the 'ea' in 'b**ea**ch' is represented. The sound is that of a long 'e' sound, characterised by the formants shown in the 8th segment in figure 3.8 and written as /i:/. The ':' represents an elongated sound.

And so, we have the resulting annotation shown in figure 3.11, where the phrase 'wreck a nice beach' is simply spread out accordingly for each of the segments we've defined, with the phonetic transcription we've created shown below that.

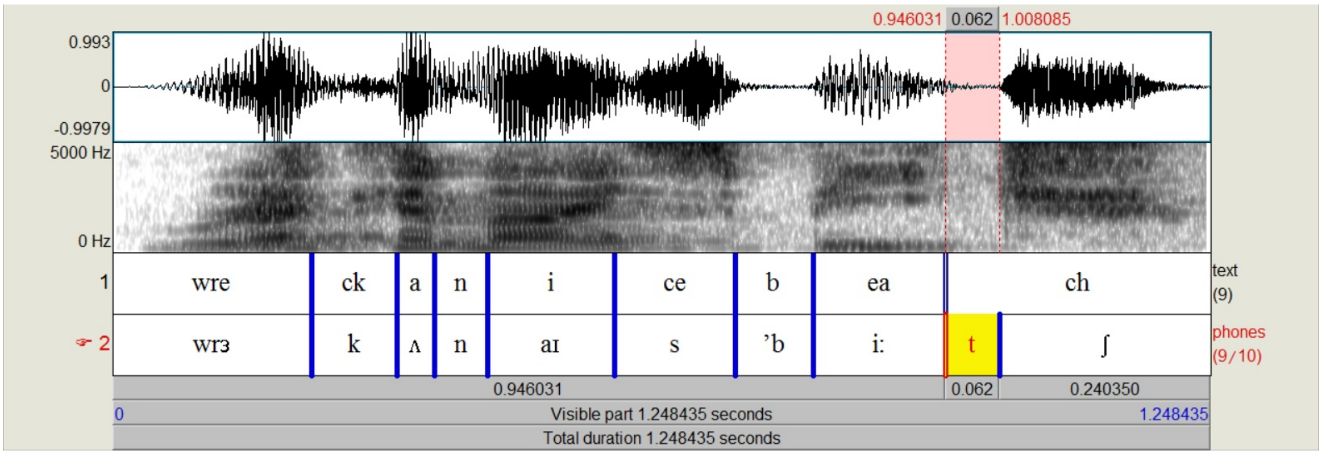
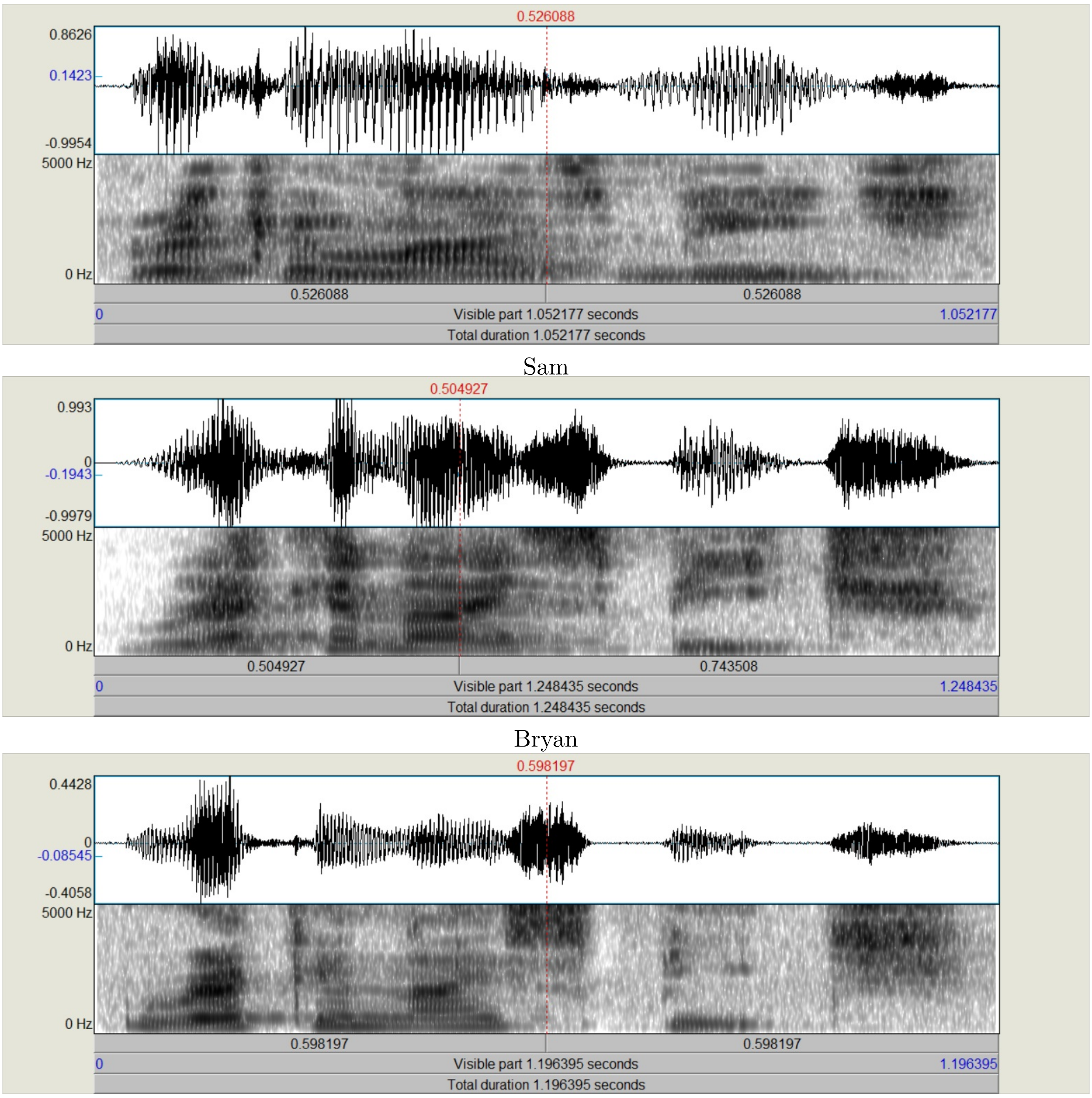


Figure 3.11: Completed annotation of the speech sounds in each section of 'wreck a nice beach'. The textual representation is shown, with a phonetic transcription below, based on the sounds in the recording.

Thus, the string of sounds for 'wreck a nice beach' would be: /rɜkʌnaɪsʹbi:tʃ/. This can be broken up as, of course: /rɜk/ /ʌ/ /naɪs/ /bi:tʃ/. If the speech recognition application made it this far to the phonetic transcription, then all that is left to do is to translate the string of speech sounds to a string of words. There will likely be more than one possible string of words that the phonetic string can be transcribed as. In this case, another possible sequence is: /rɜk/ /ʌn/ /aɪs/ /bi:tʃ/. Where the resulting words would be essentially 'wreck an ice beach'. Both are syntactically correct, though both are actually not what would be considered a 'common' thing to say in English. If, say, the former sequence of words is more common, then that would be the final transcription: 'wreck a nice beach'.

Different speakers may have different pronunciations of the same phrase, resulting in variances in transcription for the same phrase. Sometimes, though, the actual speech signals may be quite similar, and will most likely give the same transcription for the speakers. Figure 3.12 shows the acoustic signals we produced when speaking this 'wreck a nice beach' phrase. Notice the vertical striations in each spectrogram, and how they are furthest apart for Sam and closest together for Savannah. This is due to differences in pitch, where from Savannah to Bryan to Sam, the average pitch decreases from the highest to the lowest of the three.



Sam

Bryan

Savannah

Figure 3.12: A visual comparison between the acoustic signals for 'wreck a nice beach' as recorded by Bryan, Sam, and Savannah.

Some speakers may voice the 's' sound in 'nice' a little, resulting in more of a slight /z/ instead of the /s/ in /naɪs/ (this can be seen a bit in the 's' in Sam's recording, just after the red dotted line; the formant located at the bottom of the spectrogram tells us that this is slightly voiced). This may then cause the dictation software to transcribe the phrase as 'recognise speech'.

We know that one of the goals of an SR is to figure out what phones the different formants in a speech signal represent, though, this is very difficult if they look rather similar (the words sound very similar when heard). A popular example is trying to differentiate between 'recognise speech' and 'wreck a nice beach'. It is difficult for most SR systems to differentiate between the two, and the transcriptions of these are usually mixed up in some way[9]. Even just by looking at the two in figure 3.13, it is easy to see where the similarities lie, in both of the oscillograms and spectrograms. There are also a few differences.

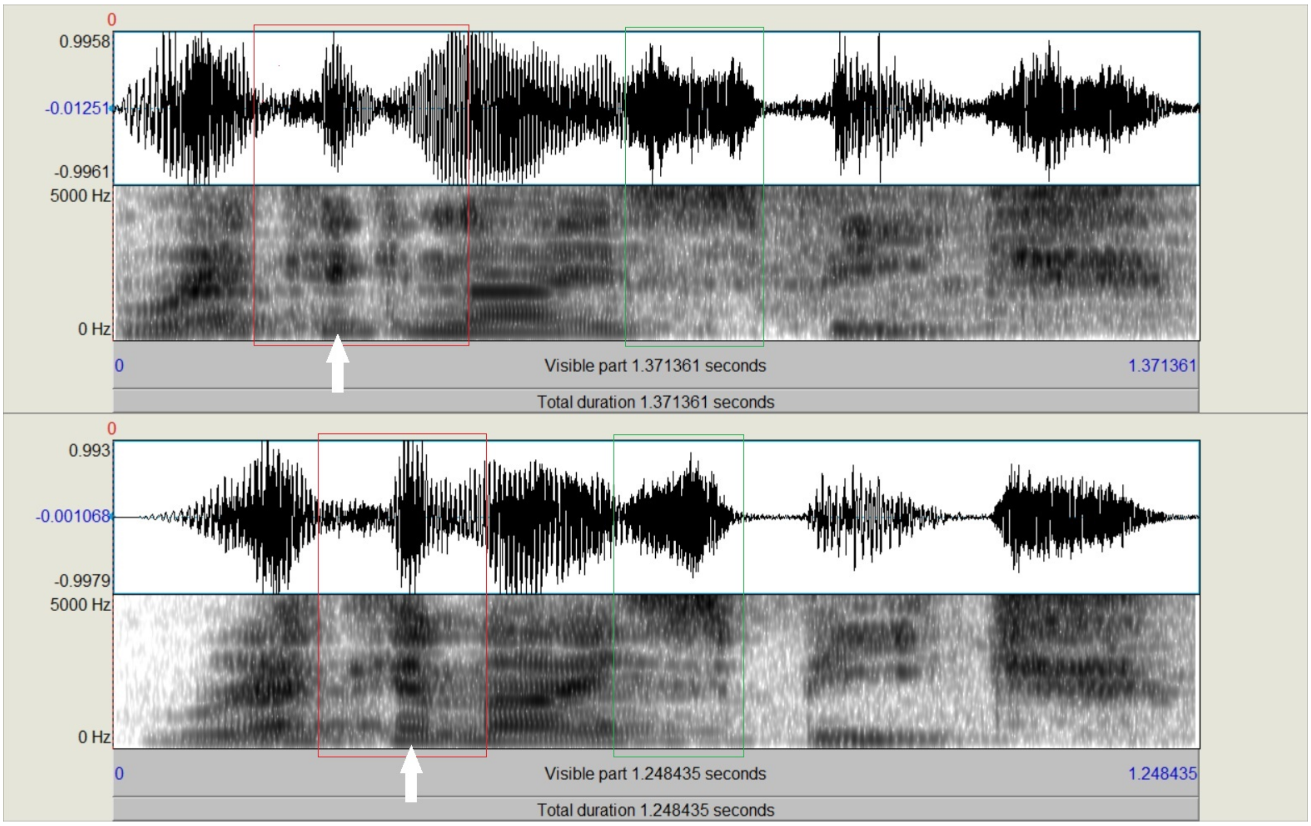


Figure 3.13: Comparison between the visualisations of the phrases 'recognise speech' (top) and 'wreck a nice beach' (bottom).

One difference is the /rɜkʌgn/ in '**recogn**ise speech' and the /rɜkʌn/ in '**wreck a n**ice beach', marked by the red boxes in fingure 3.13. The arrows signify where the /ʌ/ is in each. The part in the top acoustic signal has an extra stop (plosive) after the /ʌ/ that blends into the /n/, whereas in the bottom signal there is no extra sound there. This, however, could also be seen as a similarity, because the /g/ stop in 'reco**g**nise speech' is not so apparent, and in the recording the whole 'gn' sort of melds together (this can be seen by the movement of the lowest formant in the upper signal). The green boxes are created around the fricatives—the /z/ in the upper signal and /s/ in the lower—in the two phrases. It is expected for the fricative in 'recogni**se** speech' to be more or less voiced, with the fricative in 'wreck a ni**ce** beach' as unvoiced. Bryan, in this recording seems to have actually managed to opposite, according to the formants in the green sections. We will see how this could be transcribed in our tests in chapter 6.

[9] See section 6.2 for our tests on these phrases with the two SR applications we chose.

3.4 Summary

The process that a computer would theoretically have to go through to recognise an utterance from a linguistic perspective, is essentially:

1. Someone speaks some words, for instance, ‘wreck a nice beach’, which are recorded.
2. The speech is represented as an acoustic signal, a waveform.
3. The waveform is put through a Fourier analysis, visualised as a spectrogram.
4. The signal is broken up according to formant patterns and analysed to determine the individual phones or phonemes, resulting in a string of these sounds that represents the original words spoken, eg: /rɜkʌnaɪs′bi:tʃ/
5. The resulting string is then analysed and compared to a dictionary of transcribed speech sounds, checked with a corpus of sounds and a language model that determines the possible combinations allowed in the language.
6. The result is a set of sequences of possible combinations of words, each with a probability of being the correct sequence based on what is allowed in the language, eg: ‘wreck’ ‘a’ ‘nice’ ‘beach’ / ‘wreck’ ‘an’ ‘ice’ ‘beach’ / ‘recognise’ ‘speech’.
7. All unlikely sequences are eliminated, with the one that is most correct being chosen, eg: ‘wreck a nice beach’.

Now that we have an idea of what speech may look like, we can see how an SR might handle this input. The complete set of visualisations can be seen in Appendix C.

4 Speech-to-text

Keywords: *STT, speaker dependent/independent, acoustic model, language model, pronunciation dictionary/lexicon, speech corpus, LVCSR, Markov property, HMM, state.*

At this point we will investigate Speech-to-text technology itself. We will learn about the architecture of a typical speech-to-text system and the different ways in which it identifies and classifies the speech waveforms into words, outputting that as text.

4.1 A Typical System

In this section, we explain how an SR system works. The steps show the process of what happens during the interpretation and conversion processes. Figure 4.1 is linked to this area.

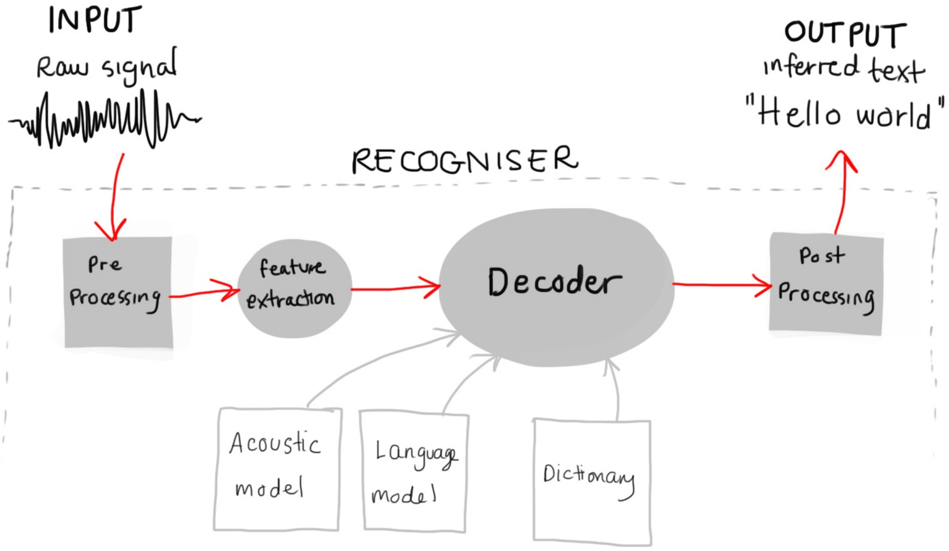


Figure 4.1: The recognition process in a typical system.

4.1.1 Pre-processing

The program's input system converts the analogue waves into a digital format. This pre-processing step starts the conversion. The sound samples are digitized and are then filtered to remove any unwanted noise. Noise reduction is performed by mathematical manipulations and algorithms that are applied to the input signal, for improvement and modification. In general, this happens by removing frequencies that interfere with the speech signal, which are registered as continuous analogue/digital frequencies. As stated by Oppenheim et. al:

> "Usually, the first step is conversion of the signal from an analog to a digital form, by sampling and then digitizing it using an analog-to-digital converter (ADC), which turns the analog signal into a stream of discrete digital values.

> Often, however, the required output signal is also analog, which requires a digital-to-analog converter (DAC). Even if this process is more complex than analog processing and has a discrete value range, the application of computational power to signal processing allows for many advantages over analog processing in many applications, such as error detection and correction in transmission as well as data compression."

4.1.2 Feature Extraction

The next step is the feature extraction after the filtering; there is a separation process, the Fourier transformation, which breaks up the signal into the frequencies that make it up. The sound is normalized or adjusted to the constant volume level. [14] People will naturally speak in a continuous manner, where the system will have to adjust and complement the speed of the sound samples on a template in the database of phones/phonemes–the dictionary.

4.1.3 Decoding

The decoding step of the system uses algorithms and models to statistically estimate the most likely spoken phrase. The input signal is divided into very small segments. The system then matches these segments to the programmed phones in the database. The system is dependent on the processing power and storage capacity, and normally only has about 20-60 different phones in the database. There are two types of programs some are designed for discrete speech while others are designed for continuous speech. Some systems prefer words being spoken individually (discrete). However, humans have the tendency to speak conversationally (continuous), with few pauses.[14]

Through various programming and techniques, speech recognition programs can recognize speech by identifying the component sounds, or individual phones. Taking the English language, there are 26 letters in the written area, and in the spoken area there are about 40 possible phones [14].

The system uses an algorithm to compare the phonemes to the words in the dictionary.[14] This is done by the statistical modelling systems. The most common ones available are Hidden Markov Models and Neural Networks. We will focus on Hidden Markov Models as they are the most common. These systems need a lot of training. quote by John Garofolo [14]:

> "These statistical systems need lots of exemplary training data to reach their optimal performance—sometimes on the order of thousands of hours of human-transcribed speech and hundreds of megabytes of text. These training data are used to create acoustic models of words, word lists, and [...] multi-word probability networks. There is some art into how one selects, compiles and prepares this training data for 'digestion' by the system and how the system models are 'tuned' to a particular application. These details can make the difference between a well-performing system and a poorly-performing system—even when using the same basic algorithm."

4.1.4 Post-processing

The post processing combines the occurrences during the pattern matching and chooses the one that achieves the highest probability. The software decides from samples, which fits best and what the system thinks the spoken word is. [19]

The program will display or perform the command required, which in this case would be the text transcribed to the document. The final outcome of the system can be altered by the user– either through the program and its capability, or by manually adjusting it.

4.2 Models

The programs and systems use methods of distinguishing between phrases and words that would fit together, by using a mathematical 'guessing' process called Hidden Markov Models (HMM). HMM will be explained in more detail in the next section, 4.2.1: Hidden Markov Models. HMMs are the most commonly used algorithms in current technology. There are usually three layers of Markov models: one to fit the sound pattern to the phonemes (the *acoustic model*), one to fit the phonemes to words (the *pronunciation dictionary*), and then finally one to fit the words to sentences (the *language model*).

4.2.1 HMM

Most ASR systems use a powerful tool called Hidden Markov models. In most cases they are used in more than one place. There is usually one that is used to figure out which phones or phonemes are used. There is another HMM that then uses these sounds to find possible words that they form. Lastly, a final HMM takes the possible words and tries to make sentences. The last HMM will be explained in more detail.

To explain hidden Markov models, we will use a common example and compare it with speech recognition systems. The reason for using this model is the fact that most spoken words are ambiguous. And some words are even pronounced in the same way, while having different spelling. Therefore a system is needed to try and create sentences. This allows to correctly dictate the speaker.

The model is used to figure out the most likely combination of words to make a reasonable sentence, given some rules about language. These rules involve the studied language. We will demonstrate this by taking the sentence:

'I will be right back'

From the perspective of the computer, most of the words in this sentence are ambiguous:

- The word 'will' could be: a name, a testament or willpower. In this case though it explains that the person, 'I', is going to do something. The computer has to figure out what the meaning of the word is, in order to be able to create the correct sentence.
- The word 'right' is pronounced the same as 'write', and has multiple meanings. To be able to choose the correct word, the system needs to understand the previous word or words.

- The word ‘back’ also has many meanings, depending on the context. It could be part of the body, a description of where something is, or description of where you are.

To make it clear, the word ‘understand’ is not entirely correct. The system uses rules and decides based on probabilities. It does not reflect or ‘think’ about what has been said.

Word Groups – In the vocabulary of the system the words are grouped based on their properties. The properties of the words are based on whether they are nouns, verbs, adjectives etc. And, in many cases, are further grouped to deal with tenses. The rules used define the efficiency of the system. Many years of research has gone into making good rules and it is a field that constantly growing. The rules are designed to people speaking normally. It is unlikely able to deal with someone saying random words that do not form sentences. The system is also unable to create words/sounds that it does not have in its dictionary. If a particular word is not in the computer’s dictionary, then the transcription for that part will most likely be wrong.

HMMs can be exemplified using an example with three states. For simplicity, the example uses chests and coloured balls, as shown in figure 4.2. The chest corresponds to word groups, and the coloured balls correspond to the words that are spoken.

There are three chests that you are unable to see. Each chest contains 10 balls. There are three different colours of balls, Red, Blue and Green. There are 10 of each type of ball.

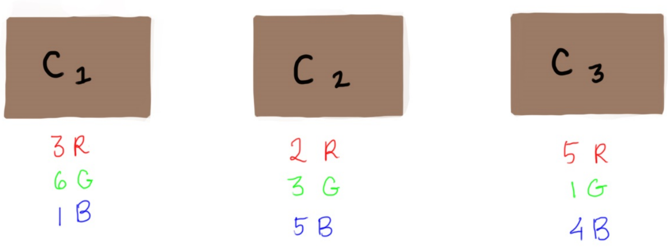


Figure 4.2: Three chests containing 10 balls each.

The idea is that someone is taking one ball at a time, from a chest unknown to you, telling you the colour. The ball is then put back. We assume that we know the likelihood for each colour of ball in each of the chests. We also assume that we know what the chance of the person changing the chest is. The likelihood for a certain colour of ball is the *output probability*. The chance of the change of chest is the *transition probability*. The picture in figure 4.3 shows what we know of the system and the probabilities involved.

Since chest number 1 contains six Green balls, then the output probability for a green ball from chest number 1 is 6/10, or 0.6.

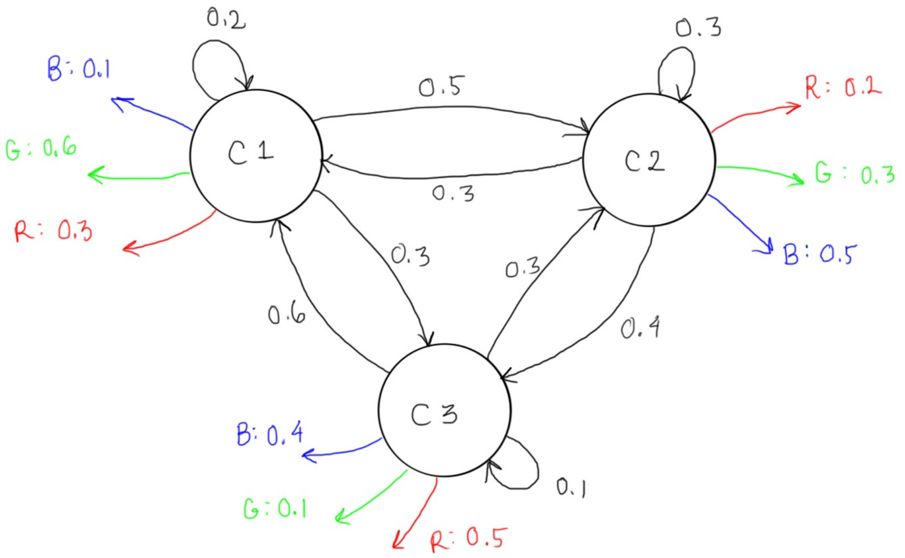


Figure 4.3: Example HMM representation for three chests containing 10 balls each.

The number of arrows can be reduced for the state diagram in figure 4.3; the output probability and the transition probability can be multiplied with each other. In figure 4.4, it now shows the probability that it will give a particular colour and then go to a particular chest.

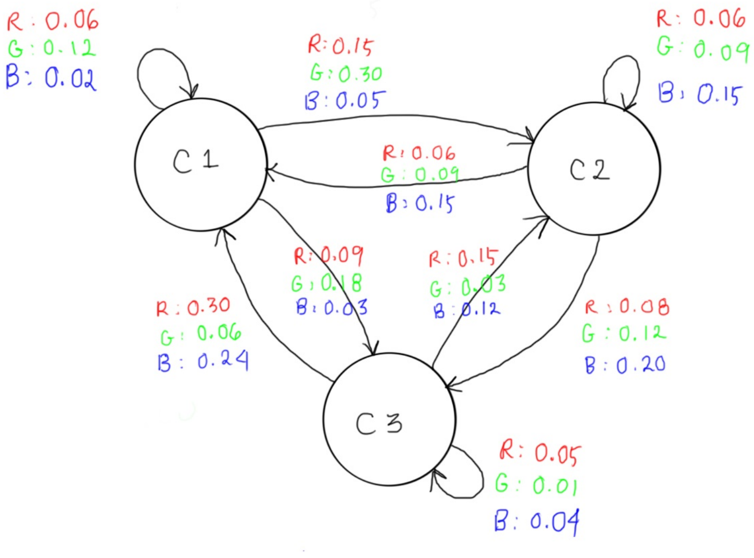


Figure 4.4: HMM representation with combined probabilities for change in states and output.

Let's say that the sequence of colours that you are given is:

'RGBBGBRR'

The output is known as the *observed sequence*. The order that the chests were chosen is the *state sequence*.

The mathematical way of describing the best sequence is

$$\hat{S} = \underset{s}{argmax} \cdot p(S \mid O)$$

Where: O = observed sequence.
S = state sequence.
$\hat{S}$ = the optimal state sequence.

which means take the state sequence with the highest probability given the observation sequence.

OBS is which colour was observed. *State* means which chest the balls came from

This table shows what information we know.

OBS	$O_1(R)$	$O_2(G)$	$O_3(B)$	$O_4(B)$	$O_5(G)$	$O_6(B)$	$O_7(R)$	$O_8(R)$
State	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8

We can only see what the result is. We cannot see which state that the observation is from.

To work with the equation we will need to expand it first

$$p(S \mid O) = p(S_1 \mid O) \cdot p(S_2 \mid S_1, O) \cdot p(S_3 \mid S_{1-2}, O) \cdots p(S_8 \mid S_{1-7}, O)$$

With this then we say that the state is affected by all previous states. But we assume that is not the case and apply what is called the *Markov assumption* (see below), giving a much simpler equation.

The *Markov Assumption* is assuming that the current state depends only on the previous state. Even if the states before the previous are known then they have no effect on the current state. The system does not ‘remember’ what has happened so can only look at the last instance and calculate from that.

$$p(S \mid O) = p(S_1 \mid O) \cdot p(S_2 \mid S_1, O) \cdot p(S_3 \mid S_2, O) \cdots p(S_8 \mid S_7, O)$$

The problem with this is then that it is very difficult to compute these percentages. Fortunately there is a Mathematical rule called Bayes theorem:

Bayes Theorem: $p(A \mid B) = \frac{p(A)p(B|A)}{p(B)}$

Bayes theorem states that there is a relationship between $p(A \mid B)$ and $p(B \mid A)$. $p(B \mid A)$ (likelihood of B given A) can be multiplied by $p(A)$ (likelihood of A without knowledge of B) and divided by $p(B)$ (likelihood of B without knowing A).

These new expressions are easier to calculate in this case.

By applying this rule we change the mathematical expression

$$p(S \mid O) = \frac{p(S)p(O \mid S)}{p(O)}$$

$p(O)$ is a constant in this case and can therefore be excluded

$$p(S \mid O) = p(S)p(O \mid S)$$

$p(S)$ is the state transition probability and can be expanded mathematically

$$p(S) = p(S_1) \cdot p(S_2 \mid S_1) \cdot p(S_3 \mid S_{1-2}) \cdots p(S_8 \mid S_{1-7})$$

Again we can apply the markov assumption.

$$p(S) = p(S_1) \cdot p(S_2 \mid S_1) \cdot p(S_3 \mid S_2) \cdots p(S_8 \mid S_7)$$

$p(O \mid S)$ can also be expanded in a similar matter.

$$p(O \mid S) = p(O_1 \mid S_{1-8}) \cdot p(O_2 \mid O_1, S_{1-8}) \cdot p(O_3 \mid O_{1-2}, S_{1-8}) \cdots (O_8 \mid O_{1-7}, S_{1-8})$$

It is then assumed that the ball that is drawn is actually only dependent on the chest that was chosen.

$$p(O \mid S) = p(O_1 \mid S_1) \cdot p(O_2 \mid S_2) \cdot p(O_3 \mid S_3) \cdots p(O_8 \mid S_8)$$

Now the two expressions can be combined.

$$p(S \mid O) = p(S_1) \cdot p(S_2 \mid S_1) \cdot p(S_3 \mid S_2) \cdots p(S_8 \mid S_7) \cdot p(O_1 \mid S_1) \cdot p(O_2 \mid S_2) \cdots p(O_8 \mid S_8)$$

There is also a start state before the first ball is taken. It is called S_0. When going from the start state to the first chest no ball will be taken. This means O_0 has only one possible outcome, it gives nothing.

After the last ball has been taken we reach an end state. In this case that state would be S_9.

We can group the Equation with the help of these two states

$$p(S) \cdot p(O \mid S) = [p(O_0 \mid S_0) \cdot p(S_1 \mid S_0)] \cdot [p(O_1 \mid S_1) \cdot p(S_2 \mid S_1)] \cdots [p(O_8 \mid S_8) \cdot p(S_9 \mid S_8)]$$

Each transition is affected by two calculations: the probability of a particular coloured ball being taken from the current chest, and the probability of which chest will next ball will be taken from.

The formula can be generalised as

$$p(O_k \mid S_k) \cdot p(S_{k+1} \mid S_k)$$

or if we use the combined probabilities from the figure showing the combined probabilities, it can be shortened to

$$p(S_k \xrightarrow{O_k} S_{k+1})$$

This is the finished formula for the example.[1]

Often the system will have multiple guesses as to which words there are so it will run the HMM multiple times to see which gives the best fit.

The purpose of the calculation is to find the best fit, but for that to happen it needs to calculate all the fits. With large vocabulary systems this can be a very difficult task.

[1] For more information on the math used here, refer to this lecture https://www.youtube.com/watch?v=mqI468LMThg

The amount of calculations is decided by the number of possible states (word groups) to the power of the number of observations (length of the sentence). So if you have 200 groups of words and you have a sentence that is 15 words long then you have 200^{15} different calculations . The longer the sentence the more calculations. This means there is a lot of processing time. There are ways to deal with this but will not be covered in this report. If you would like to know about it read about Viterbi algorithms.[2]

4.3 Summary

To summarize what has been employed and introduced in this chapter, we will go through the steps briefly. We will use the diagram in figure 4.5 again to illustrate the process:

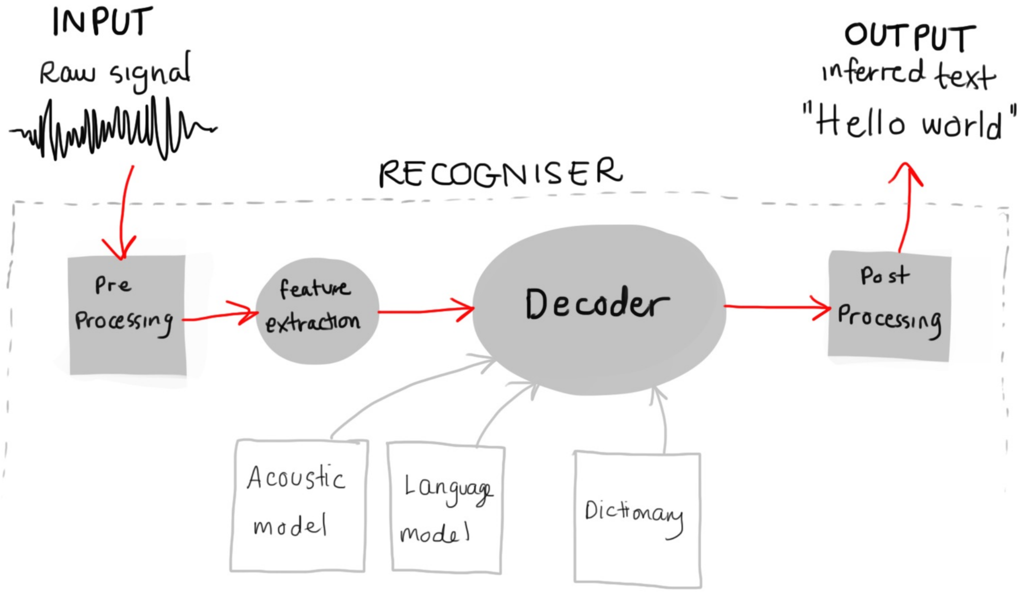


Figure 4.5

- Speech → Analogue transmission of sound signal.
- Pre-processing: Conversion of analogue waves to digital data. Filters and noise reduction used.
- Feature extraction: Sound is normalised, volume is adjusted, and the signal is broken down into pieces.
- Decoder: The decoder uses Hidden Markov Models to evaluate a good fit as to what the speaker said, first it matches phones and phonemes to each slice of the signal based on the features of each piece. Then it matches the phones/phonemes to words. Finally it creates sentences from the words.
- Post-processing: The possible sentences are analysed and the most likely is chosen
- Output: the transcription is then finally written down into the text field that the user has selected.

[2] http://en.wikipedia.org/wiki/Viterbi_algorithm

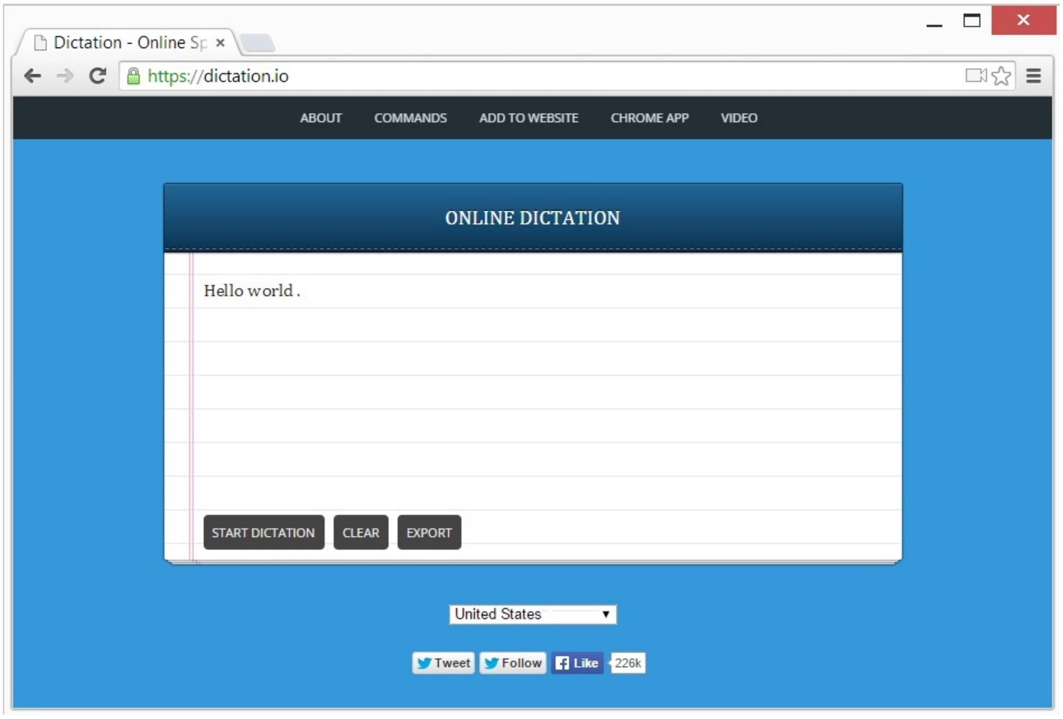
The mechanics behind STT are explained. The following chapter will look into the systems chosen for the project. With the understanding of how STT works, we can apply this theory in practice.

5 Applications of Speech Recognition

The two SR systems we chose are both designed to recognise large vocabulary continuous speech. These are Windows Speech Recognition and Google Chrome's Dictation. The latter utilizes Google's speech recognition engine for its Voice Search feature. Both are only usable with the Google Chrome browser, whereas Windows Speech Recognition is found on any PC with Windows Vista, 7, or 8/8.1 installed.

5.1 Google Chrome's Online Dictation

Google Chrome's Dictation application (we will just call it Dictation from now on) is a web-based dictation application utilising ASR to allow users to dictate their emails, essays, and such. It can be done by going to the webpage in Chrome[1] and pressing the 'Start Dictation' button to transcribe whatever it is that should be written down. Dictation is purely for dictation purposes, as the name suggests, and does not allow for any commands outside of the few it offers. It utilises the built-in speech recognition engine of Google Chrome for the recognition process, with support for many languages[2].



Hello world .

Figure 5.1: Google Chrome's web-based Dictation application, with dictated words 'Hello world'.

Dictation is able to recognise a continuous flow of speech, with an additional small set of editing commands: 'new paragraph', 'comma', 'full stop', and 'question mark'. Because Dictation is actually able to transcribe continuous speech from many different speakers with varying languages, even with some errors, it can be classified as a

[1] https://dictation.io/

[2] Including Arabic, Chinese, French, German, Indonesian, Italian, Malay, and Spanish

speaker-independent system. Additionally, it should even be a large vocabulary system, considering the variety of speakers and their most likely varying choices of words, and the fact that it can support many different languages and dialects, though we cannot realistically test this. Thus, this type of application is classified as an LVCSR. When using this application, we noticed that it does seem to favour certain words over others that may be less common, or mistake two separate words for a single words that is a combination of the two.

According to the writers at Wired.com, Google has been using a neural network model to improve their SR software in the Android OS [21], though we do not explicitly know if they use the same process for their non-mobile-based SR systems.

5.2 Windows Speech Recognition

Windows Speech Recognition, referred to as WSR hereafter, is a feature that allows for control of one's PC with their voice. Users are able to interact with and navigate the Desktop, file system, and various applications with the appropriate voice commands[3]. There is also the ability to dictate words into word-processing programs, such as Notepad or Microsoft Word, as well as fill out forms or text fields online or in pdfs. There are also commands to add punctuation, correct, and edit the transcribed text that can be used during the dictation[4]. The program itself supports a handful of languages[5] as well.

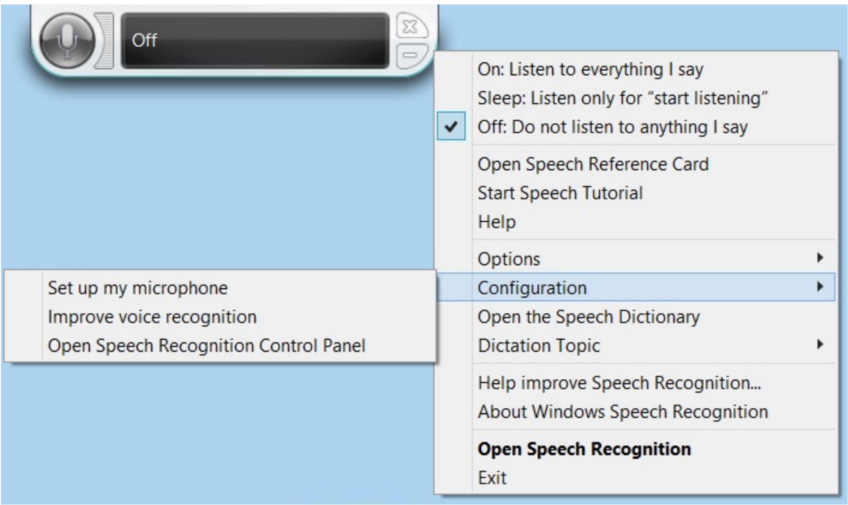

Off
On: Listen to everything I say
Sleep: Listen only for "start listening"
✓ Off: Do not listen to anything I say
Open Speech Reference Card
Start Speech Tutorial
Help
Options
Configuration
Open the Speech Dictionary
Dictation Topic
Help improve Speech Recognition...
About Windows Speech Recognition
Open Speech Recognition
Exit
Set up my microphone
Improve voice recognition
Open Speech Recognition Control Panel

Figure 5.2: Windows Speech Recognition.

To use WSR, one has to search for it (using either the search box in the Start menu in Windows Vista and 7, or via the search charm in Windows 8/8.1) and select the WSR. A small bar would appear, normally at the top of the screen, with a button with a microphone on it. Right-clicking anywhere on the bar, there appears multiple options, including the option to train the system to the user in order to better recognise their speech with 'Improve voice recognition' (figure 5.2). When first using WSR, it is able to transcribe the given speech fairly well, though there are still many errors that can

[3] http://windows.microsoft.com/en-us/windows/common-speech-recognition-commands#1TC=windows-7

[4] http://windows.microsoft.com/en-us/windows/dictate-text-speech-recognition#1TC=windows-7

[5] English, French, German, Japanese, Spanish, and Simplified and Traditional Chinese.

occur if the speaker is not speaking clearly or articulating. WSR is a speaker-dependent system, even though it is still able to recognise speech without training, because it must be trained to the speaker in order to achieve its highest accuracy and efficiency. It can handle short commands, as well as continuous speech for dictation, therefore we will classify this system as an LVCSR.

While we are not sure exactly what acoustic models WSR uses for Windows PCs, a video[6] from 2012 by Microsoft Research of a speech recognition demonstration by the Chief Researcher Rich Rashid suggests that Microsoft might be utilising deep neural networks for their SR systems, at least in their translation technology.

[6] https://www.youtube.com/watch?v=Nu-nlQqFCKg

6 Tests

By conducting a collection of small tests on our selected applications, we would like to determine what transcriptions they give, and how accurate they are. Additionally, we hope to gain an idea of how they work, because we do not know exactly what models these applications use. We test how each program performs when given isolated words, single phrases and sentences, as well as longer texts. The last part, the dictation of longer texts, is done in a relatively calm setting, where there is not so much superfluous noise. All of the rest are done in a silent setting, so that we may see the effects of the speech itself, rather than the effects of the background noises. All tests were used with the same hardware: simply, a Lenovo y510p ideapad running the Windows 8.1 operating system, and its built-in microphone (most users are not wandering about with headsets). The speakers are the two males and one female that make up our group, Bryan, Sam, and Savannah. We'll record during the dictation as well.

6.1 Isolated Words

How does the application perform when given speech in the form of single words? We have selected various words from the English language that have a sort of relation to each other, listed below, that are to be spoken into the two systems:

zip	sip		gnat	bad
zap	sap		not	bade
			note	bead
fat	vat		newt	bed
			knit	bid
cat	hat		neat	bode
pat	bat	mat	net	

The pairs of words in the first two columns, under 'zip' and 'sip' and including 'mat', were chosen because they are very similar to each other, aside from only once difference within each pair. These are kown as 'minimal pairs'. With 'fat' and 'vat', for example, we have the unvoiced 'f' in the first word, and the voiced 'f'—the 'v'—in the second, with the place of articulation being the same for both. There is a similar pattern for the rest of those pairs, wherein the last pair the place of articulation is also the same for each, though the beginning consonant is aspirated ('p'), unvoiced ('b'), and voiced ('m').

The last two columns consist of words that are pronounced the same way, except for a difference in the vowels in the middle. So, for instance, with 'bid' and 'bode', notice that that the start and finish of each word is the same when pronounced, and the only difference is the middle, the vowels in between. All of these are known as minimal pair words, in which there is only one step or difference from one word to the other.

Hypotheses: Based on what we have learned from our investigation into the theory behind STT, we can conclude that systems that utilise HMMs will be not be very good at recognising individual words because HMMs try and predict the sentences given the most likely combination of words. If the applications that we chose are implementing HMMs, then we expect for them to have great difficulty in transcribing the words above. It is possible that the applications will even try to interpret the words as being part of a phrase or sentence and transcribe them in that way.

Methods: In order to test how well the applications are able to differentiate between the minimal pairs, we will read each word in a normal voice, with no extra enunciation—though with no mumbling either. In between each word, we will place some sort of punctuation so that we can separate each word to see how the application has interpreted the words. For the last two columns we will do the same, although we may try to just read down the list as well to see how the systems cope without punctuation. If we see that the application has completely misinterpreted the words, we will read them out again, just to check if that is really what it 'believes' we have said.

Results:
As expected, the results of the individually spoken words were not altogether accurate, though the applications did manage to get some of the words correct if we really tried to speak clearly.[1]

When saying the 'zip'–'zap' pairs, for instance, Dictation often transcribed these with both words in each pair starting with a 'z' for each speaker:

Speaking: '*Zip, comma, sip. Full stop*'
Transcription: `Zip, zip .`

'*Zap, comma, sap. Full stop*'
`Zap, sap .`

'*Zip, sip. New paragraph*'
`Zip zip`

'*Zap, sap. New paragraph*'
`Zap zap`

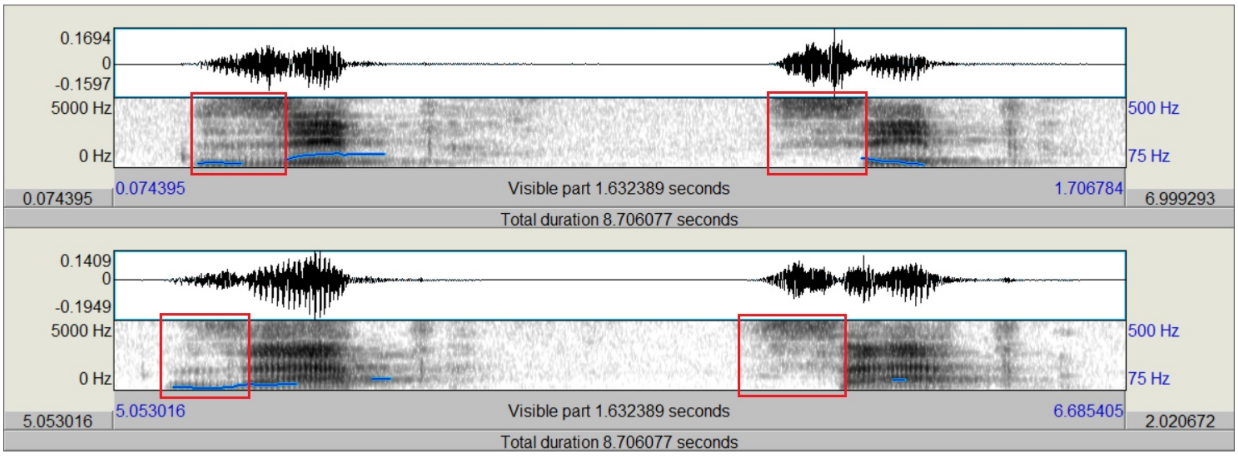


Figure 6.1: In this visualisation of the signal for '`zip`' (top left) and '`sip`' (top right), there is the absence of the formant at the bottom indicating a voiced sound for '`sip`', whereas it is shown for '`zip`'. Similarly for '`zap`' (bottom left) and '`sap`' (bottom right) just below that.

It is possible that, when speaking the words, the speakers have voiced the 's's, resulting the the transcription being with a 'z' instead of 's'. Looking at one of the recordings

[1] Refer to the first half of Appendix D for the complete results of this part of the test.

shown in figure 6.1, however, there is no indication of a voiced 's'. This looks similar for the other speakers as well. Because it does not seem to be the speaker that is causing this—and there is no extra noise, nor are there any apparent problems with the equipment—it must be the result of the software, possibly deciding for some reason that the phoneme is a `/z/` instead of `/s/`. While we are not certain of what the actual reason is, a possibility is that this transcription had a greater probability than the correct transcription. Another could be that the reference for the phone in the lexicon gave `/z/` instead of `/s/` for 'sip' based on possible pronunciations, as sometimes people can pronounce 'z's with little to no voicing.

The Windows application gave more errors in the transcription of the individual words than for Dictation, and only on a few instances did it give a reasonably close transcription, eg.:

'*Zip, comma, sip, comma*'
`Zip, zip,`

'*zip, sip. Newline*'
`As in saint`

'*Zap, sap. Newline*'
`Zzap so`

'*nat, not, note, newt, knit, neat, net. Fullstop*'
`Nats not know what you next need Nats.`

'*cat, comma, hat. Newline*'
`Cats, at`

We also believed that the applications would attempt to 'make sense' of what was spoken by generating a plausible sentence, which can be seen in some of the results above for WSR. This did not occur as much for Dictation, though for that application it did misinterpret the commands given on occasion, and put them in directly or as something similar in the transcription, for instance:

'*Pat, comma, bat, comma, mat. New Paragraph*'
`Pet, but, met new cars`

The errors in the transcription itself here seem to actually be the result of the accent, where in the recording the words actually have more of a short 'e' sound, rather than the short 'a'.

6.2 Single Phrases and Sentences

A couple of common words in the English language are: he, she, and that.

We test whether making slight changes in a particular word will cause a minimal or large change in the transcription of the sentence. For instance, 'he' and 'she'.

Why did she day that?
Why did he say that?
What was that he said?
What was that she said?

Would the applications still transcribe a word correctly if it is used in different contexts? We take the word 'that' and formulate a few different sentences containing the word at the beginning, middle, and end of the sentences.

The sentences are:

What is that?
Could you repeat that?
That's not what I said.
That is quite the problem.
That is groovy.
I thought that was right.
People don't say that anymore.

What about similar phrases? It is often difficult for speech recognition applications to distinguish between phrases that sound similar. We want to test if these two applications will be able to correctly distinguish the differences between the following phrases:

Recognise speech.
Wreck a nice beach.
Mechanise peach.

Hypotheses: The applications should have little problem in general with the first two groups of sentences, as they are normal things to say—save for 'that is groovy'. We do believe, however, that there will still be a few errors in the transcriptions for 'he' and 'she', as they are quite similar to each other. We are curious how the word 'that' will be transcribed when places in different contexts. It is possible that it will be transcribed most correctly at the beginning and end of the sentence because there are no other words on one side that can alter the sound of the word as much. The last three phrases are unusual, and so might cause a problem for the application unless they are articulated very concisely. It is possible that the first phrase is more plausible or common than the other two. In the case of an application that implements HMMs, it may be that the other two phrases will be transcribed as 'recognise speech' because they sound almost identical and this phrase may be the more popular of the three. In any case, we expect to see that two of the similar phrases be transcribed as the third.

Methods: Just as with the isolated words, we are to speak normally, placing punctuation in between each phrase so as to separate them. If the application keeps giving an incorrect transcription, we can try to speak more clearly, if that is what it takes for the application to generate an accurate transcription.

Results:
For the 'he' 'she' pairs of sentences, Dictation was able to transcribe them almost completely correctly, while WSR had more noticeable problems, for example, these errors from Sam's dictation:

'*What was that he said, question mark.*' (spoken twice)

```
One is that his own?  What was there isn't?
```

'*What was that she said, question mark.*' (spoken twice)

```
Was this reason?  What was that research?
```

And these from Savannah's:

'*Why did he say that, question mark.*'

```
Where did he see in the?
```

'*What was that she said, question mark.*'

```
What was that she's an?
```

Bryan's transcriptions were the most accurate of the group for WSR. If we look only at the results, they can suggest that there is a difference accent or in clarity in speech, perhaps pitch, that has affected the accuracy of the transcriptions. Now, each member has a different accent, and each normally speaks within a different octave than the other two. It could be, since Bryan's voice averages around the middle octave of the group, and his were the most accurate transcriptions, that it gave the best conditions for the default parameters WSR has set before training. There is also the accent. Bryan may have the clearest accent or way of speaking, whereas Savannah, for instance, has a different pronunciation of vowels, and does not always pronounce the words clearly or precisely. This seems to be more of an effect induced by the speaker, not so much the software itself.

We also dictated sentences with the word '`that`' in various positions within the phrase. The results of this part show that, for this particular word, at least, there was no real difference in accuracy of the transcription when the word was at the start, middle, or end of a phrase—a phrase/sentence that is still grammatically correct and not necessarily obscure. However, there were also errors related to pronunciation, based on the recordings and resulting transcriptions. Overall, there was no noticeable effect of placing a particular word in different positions in sentences. There was a particular instance where the word '`groovy`' was misinterpreted:

> '*People don't say that anymore, full stop . That was groovy, full stop.*' (from Dictation)
> `People don't say that anymore .  That was ruby.`
>
> '*I thought that as right, full stop. That was groovy, full stop.*' (from WSR)
> `I thought that was right.  That was truly.`

This could just be the result of the word having a low probability of appearing in that particular sentence, and so both applications chose a similar word with higher probability.

With the similar phrases, the phrase '`recognise speech`' was almost always transcribed correctly by both applications for each group member. With the only slightly incorrect transcriptions being:

> '*Recognise speech*'
> `Recognized beach`
>
> '*Recognise speech*'
> `Recognise beach`

both from WSR, as spoken by Sam. In the recording, both of these actually sound remarkably like '`wreck a nice beach`'. Because '`wreck a nice`' has nearly identical formant patterns to '`recognise`', it is probable that WSR determined that '`recognise`' was the more likely word to match the recorded speech it was given. The '`beach`' in both results there were most likely produced due to the phonetic transcription determined from the formants that represent the word. It is usually the pattern and movement of the formants in the preceding segment that determines what the consonant is. Consonants that involve a complete blockage of air flow through the vocal system, such as the 'p' and 'b', normally appear as relatively 'blank' spots in the spectrogram, as shown in the highlighted segment of figure 6.2.

This similarity may be the cause of the phrase being transcribed in this way. As for the other two, '`wreck a nice beach`' and '`mechanise peach`', these two are rather uncommon phrases to utter, and may be precisely the reason for them being transcribed as

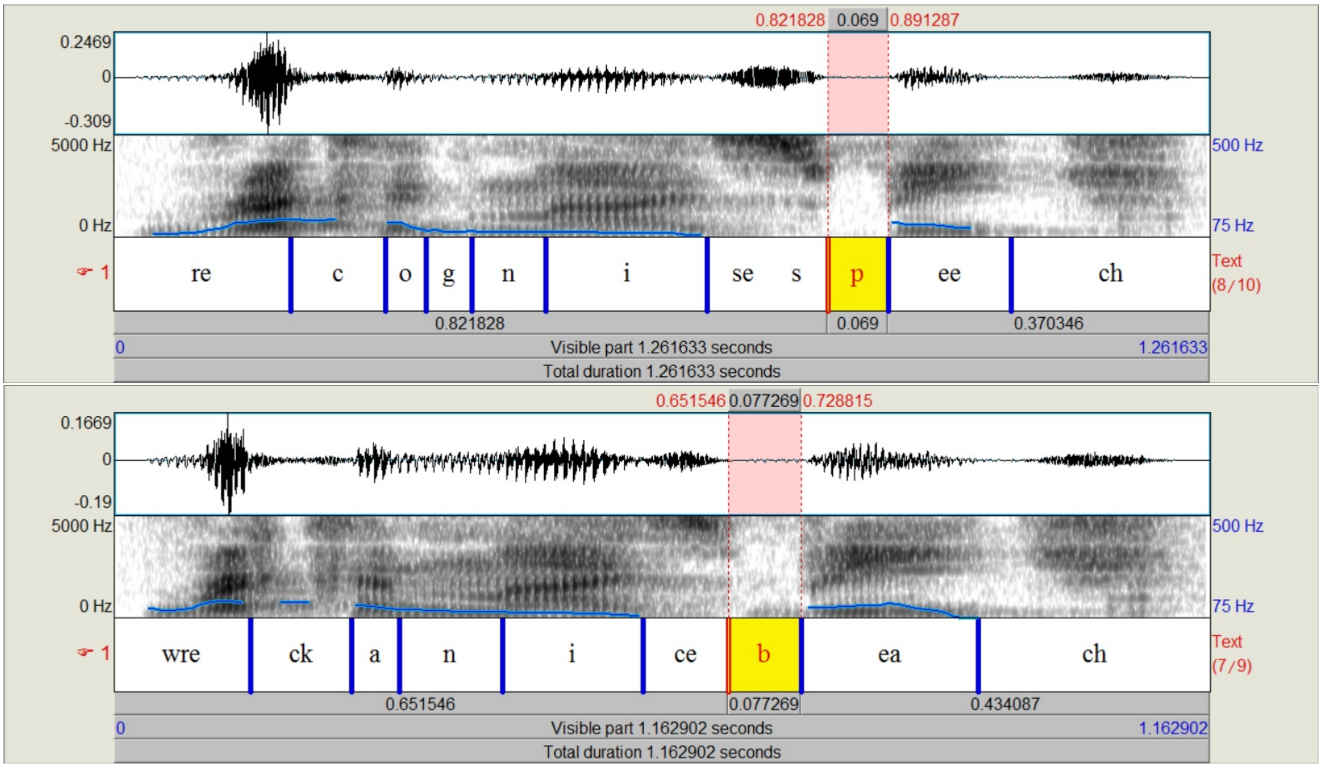


Figure 6.2: Sam's recording of 'recognise speech' (top), and 'wreck a nice beach' (bottom). The highlighted segments show examples of plosive consonants, in this case 'p' and 'b'.

'recognise speech' when speaking normally. It is also very interesting how for 'wreck a nice beach', when dictating, it was once transcribed as both:

```
Make a nice speech Making a speech
```

The Dictation application once almost transcribed 'mechanise peach' correctly, but it decided against it, most likely due to differing statistical probabilities. Similarly with transcribing 'w' as 'm'. The fact that this phrase is not something one would normally say might just be the reason for the switch from 'w' to 'm', and thus 'wreck a nice beach' to 'make a nice speech'.

6.3 Dictating Longer Texts

We also want to try dictating longer texts, roughly two paragraphs, into each application. These are a paragraph from *The Hitchhiker's Guide to the Galaxy* [1] and *Catch-22* [16].

Hypotheses: We believe that if the applications make a mistake, then it is likely that they will make a mistake in the rest of the sentence, or that the mistakes will arise in groups, where whole parts of sentences will be transcribed incorrectly. We also expect that the applications will perform best for this part of the test because they are designed for this purpose: to transcribe continuous speech. It is likely that Dictation would perform better than WSR, as it is designed to be as accurate regardless of the speaker, whereas WSR depends on the training it gets from a particular speaker in order to reach the same level of accuracy or better. We believe, of course, that the trained WSR will perform better than the untrained WSR.

Methods: We are to read the excerpts in a relatively normal and clear manner, without any extra articulation, using the commands available to insert the appropriate punctuation. We will then train the WSR application to Savannah's voice for a couple of hours (via the provided training feature and error checking) and then have her read the excerpts again using WSR.

Results:
We've noticed that Dictation has cut off part of the beginning of the transcriptions. We have found out that this is due to the processing/preparation that it undergoes after pressing the 'start dictation' button before it can be ready for the dictation. We were not aware of this for most of this part of the test, though later on we waited until it appeared to be ready. This is also a good example of some of the difficulties with processing time that SR applications have, resulting in delayed or slow response times.

Overall, the transcriptions were not perfect, but largely resemble the original text. If we were dictating a letter to someone, we would have to go back in and edit the mistakes. It is possible to do this via voice commands with WSR, though not with Dictation. The transcriptions for this part presented at the ends of appendices D and E do not include this sort of editing. We have read the texts without going back to make changes in order to show the 'raw' results of the dictation. The transcription given by the relatively trained WSR application is also without these corrections, though correcting errors was done during the training period as part of training the application.

We mentioned that the transcriptions 'largely resemble' the original text. For example, consider the following text:

> "I'm a scientist and I know what constitutes proof. But the reason I call myself by my childhood name is to remind myself that a scientist must also be absolutely like a child. If he sees a thing, he must say that he sees it, whether it was what he thought he was going to see or not. See first, think later, then test. But always see first. Otherwise you will only see what you were expecting. Most scientists forget that." [1]

The resulting transcriptions from our reading this into either application contain many errors, but we can see that the words each application decided upon may be similar to the original words—or at least the words that we've spoken; the errors could of course come from mispronunciation or accent. The transcribed words may sound similar to the words in the excerpt if one was to speak them aloud.

For Dictation, the resulting transcriptions were actually okay, with occasional error blocks.[2] Given the excerpt above, the results of reading it into Dictation are the following:

> `I'm a scientist and I know what constitutes proof but the reason I call myself by my child who name is to remind myself that a scientist must also be absolutely like a child. If he sees a thing, he must say that he sees it, what date was., think later, dentist pulled off but always see fast. Otherwise you will only see what you're expecting. Most scientists to get that.`

> `I'm a scientist and I know what constitutes proof. But the reason I call myself by my childhood name is to remind myself that the signs also be absolutely like a child. If you see this thing, he`

[2] All of the results for dictating the excerpts with Dictation can be found towards the end of Appendix D.

> `must see that he sees it, what they'd was what he thought he was going to see you're not. See first think later, then test. But always see first. Otherwise you will only see what you're expecting. Most scientists forget that.`
>
> `[...]`[3] `What constitutes proof. But the reason call myself on my child his name is to remind myself that a scientist must also be like a child. She sees a thing, you must see that he sees it, whether it is going to see or not think later then test. Otherwise you only see what you were expecting. Was trying to forget that`

These errors seem to consist of a whole sections where multiple words are transcribed incorrectly, as well as just single-word errors. This partly coincides with what we hypothesised, where we predicted that whole chunks of a sentence would be affected by a single error. However, according to our results, this is not always the case.

For WSR, the result was hardly better. The untrained application presented a few more errors than Dictation; there are many incorrect words, and whole sections that only have a small resemblance to the original excerpts. For example, with the excerpt we presented on page 44, the result was:

> `I am as scientist and who would constitute proof. But the reason I call myself to my children mean is to remind myself that scientists must also be absolutely like a child. That he sees a thing, Siemens said he sees it, whether it was what he thought he was going to see or not. Seafirst, simply, and test came early Seafirst. Otherwise you only see what you were expecting. Most scientists forgot that could`

Though, after some tries at training WSR, the overall transcription results improved. We trained a WSR system to Savannah's voice, and the improvements between the untrained reading and the trained reading were quite large, even though there were still some errors (these were at words that are usually not as popular, or words, when pronounced together, confuse the system into transcribing them incorrectly). The following is the transcription of the same excerpt with the trained WSR by the voice it's trained to:

> `I'm a scientist and I know what constitutes proof. But the reason I call myself by my childhood name is to remind myself that a scientist was also be absolutely like a child. If he sees a thing, he must say that he sees it, whether it was what he thought he was going to see or not. Seafirst, think later, then test. But always see first. Otherwise you only see what you are expecting. Most scientists forget that.`

There were only a few errors: the '`was`' in place of 'must' in the second sentence, '`Seafirst`' instead of 'see first', and the missing 'will' from '`you only see`' and '`are`' in place of 'were' in the second to last sentence. This is a big improvement from before, where there were strings of errors instead of the isolated errors shown here.

We then wanted to see how using a trained WSR for someone else's voice would affect the transcription. Below is a reading of the excerpt from *The Hitchhikers Guide* by one of the other group members into the trained WSR, and the results appear to be worse than that of the untrained WSR, with even more mistakes made during transcription:

[3] This is actually where Dictation did not catch the beginning of the dictation.

```
Imus scientist and I know what constitutes proof.  of the reason
I call myself I was halted name is to remind myself and scientists
must also be obsolete me like a child.  if he sees the reigning,
king city that he sees that, whether it was what he thought he was
going to C1, OC frost think we can, then test, the woolly C frost.
otherwise you'll only see what you were expecting.  of scientists
forget that three com
```

As a result of adjusting the paramaters for the phones to Savannah's voice, it has made it much more difficult for the application to accurately transcribe the words that were spoken by the other speaker. It can be compared to adjusting one's computer screen to the low lighting of night-time outside, and then bringing the computer inside to a well-lit room. The screen would seem much too dark upon entering the room from the relative darkness of outside, making it difficult to see anything shown on the screen.

A continuation of this discussion is carried out in the next chapter, where we also attempt to answer the two discussion questions posed at the start, in chapter 2: Preliminaries.

7 Discussion

7.1 Theory in Practice

In general we can see that spoken sentences and phrases give more accurate results than that of individual words. This is not unexpected as we assumed that the applications utilise Hidden Markov Models. The individual words do not fit into what a speaker is expected to normally say.

There is a difference between the speakers. This can be seen on the test results. There is also a big difference when the speaker articulates in different ways—this is also not surprising. The more clearly the person speaks the more likely that the system assigns the correct phonemes, as they will be easier to identify. If the correct phoneme is found, then the system is more likely to write the correct phrase. Since we didn't use any kind of headset during the tests, the quality of the input is questionable. This means that a lot of the mistakes may have been caused by bad quality input.

We could make some other general inferences as well. The applications that we used are definitely large-vocabulary, as they must be in order to be successful dictating applications. We've noted before that we believe Dictation was speaker-independent, and WSR as speaker-dependent. This can be confirmed through the tests we have run, as Dictation was able to handle a lot, even though it only had a handful of editing commands. On the other hand, WSR had a larger library of commands that allow the user to edit and add various punctuation to their documents. However, WSR is not as efficient or accurate when first used, since it needed training to achieve the same accuracy as a very good speaker-independent SR system. We do not think that we can classify Dictation as a 'very good' speech-to-text system, as there was still a very high word error rate for our transcriptions.

We would like to try and infer what models these two applications use, though, in order to have a good idea, we would need to run more robust tests that really pinpoint the different aspects of the models. A safe guess would be that both use HMMs, as those are most popular among speech recognition systems today. Though, WSR has the training feature, and such is common with systems that use either a neural network, or a combination of HMMs and another model. We've also noted that Microsoft has introduced neural networks into some of their products, as a way of increasing accuracy and efficiency, so it is actually possible that the WSR application we have used could have at least an HMM and neural network-type combination. We would need to have this sort of documentation in order to know what models are actually used, though for the future we could run more robust tests in order to make better inferences.

Another expansion we could add to the future 'todo' list is to better duplicate the excerpt test for both the Dictation and WSR applications, since, as of now, we have not made a reading for each member for WSR, as we have for Dictation. If we were to redo this test, we would do more readings and for each group member, before and after training the WSR application. There is no recording of the dictation of these longer texts, and we wish that we would have done that so that we would be able to see what the reasons were for the errors we got. If we had recorded both the first dictation, and the dictation of the transcription, then we would be able to see what the acoustic signal would actually look like, had we actually said what was in the original transcriptions.

We would also extend the time of training WSR, and even have a version trained for each member of the group. In addition to this, it would be a good idea to repeat the words and phrases tests with WSR after it has been trained, both with the person it is trained to and with the other two to which it is not trained. This would give an even better picture of the extent to which training the system can improve its functionality in the areas that we have tested.

7.2 Can Computers Recognise Speech?

The purpose to this project was to study and evaluate computer speech recognition to see if it can take spoken words and create text, and how it does this. The tests show that it is possible to a degree to do this. It is known what individual sounds look like and, because of that, it is easy to create a computer than can 'talk'. The problem is that not everyone says these sounds identically and the sounds can be used in a variety of different ways. The machine still has to figure out which words each sound is from, how long the words is and what meaning of the word represents. An analogy is 'getting a computer to create speech is like squeezing toothpaste out of a tube of toothpaste'. Getting a computer to understand speech is like trying to put it back in again. Having studied how HMMs work we can see that many things have been learned in the last 40 years of research. But it is far from the perfect system. In most cases the system is too inaccurate and slow to be something that people would want to use. People would much prefer using a keyboard or get someone else to do so.

We have looked into what is understood by speech. How sound is created and what qualities speech has. These qualities are well understood and help create better systems. Everyone speaks slightly differently. This makes it very hard to make rules for a machine that allow it to understand speech from more than one person. Even if it is only one person, there is still the fact that each utterance can be assigned to different words depending on the context.

It is very difficult to make a good SR because of all the things that can cause complications in the recognition process. Among others, there are:

- The different ways in which people speak, like their accents, speed of speech, intensity, even differences in pitch.
- Whether or not they speak clearly, mumble, or otherwise.
- Whether or not they are sick, or have a disability or speech impediment.
- Environmental surroundings and location of the speaker.
- And, of course, other people that are speaking nearby.

The extra noises that occur during the recognition process can show up in the speech signal and alter the classification process, where the computer has to say 'this is this phoneme, and this bit must be that phoneme...', and so on. Many have used various SR applications found in mobiles, usually as 'personal assistants' like Siri or Cortana, and have a rather good idea of how such factors can impede the recognition process. When using the assistant found on the newer Samsung Galaxy smartphones outdoors, for example, when it is even just slightly windy, it is extremely difficult for the application to recognise what is being said. In order to get the most accurate and efficient response, one must be located inside in a quiet room. Though, this is not always what we can do; it is impractical, always having to excuse yourself to a quiet room just to use the personal assistant on your phone that is supposed to be able to handle these kinds of situations. So, when would it actually be practical?

There are the kinds of situations where one hasn't any use of their hands; they may either have their hands full or are disabled in some capacity. In this case, yes, it would be rather useful to be able to do the sort of thing that would usually require the availability or use of the hands. Or, even cases where doctors need to take notes during examinations or surgeries. It seems that they must perform a part of the examination, remember the information that they have just gathered, pause the examination, turn to their notes and proceed to write down this information, and then turn back and resume the procedure.

In some cases there is a person that takes the notes for them, maybe in order to work around this. In some instances like this where examinations of some sort are taking place, there can be an assistant present that takes notes of what the examiner is telling them. In these cases, they must speak clearly in order for the person to understand what is being said, when these words are most likely being spoken into the patient (in the direction of the patient, much like speaking into one's pillow where it is hard for others to understand what is being said) and not out and to the person taking the notes. Here is where the examiner would probably already have to speak clearly, which is advantageous since most speech recognition technology requires relatively clearly-spoken speech in order to get the most out of the technology.

8 Conclusion

Though the domain of speech recognition is vast, this project is an attempt gain a better understanding of the different areas of speech recognition, specifically that of speech-to-text software. Through our research, we identify, illustrate and evaluate how speech-to-text enables the first step in allowing computers to effectively interact with humans: the ability to recognise speech. We see that speech recognition also solves the issue in enabling a hands-free aspect to interacting with machines. The development of the technology has proved useful in people's daily lives, enabling dictation methods that reduce the amount of work performed by the user. Systems are developed to accommodate the users, where the two types, speaker-dependent and independent are available. There are limitations that determine the progress of the systems. These limitations can be seen in the real world with various applications of SR, where the software and hardware components are determining factors. Using the speech to text programs on our devices has also confirmed that there are numerous factors to be taken into account. By comparing the results of what various speech signals look like using Praat, we understand the difficulty of programming systems to interpret these sound waves, in our case under a controlled environment. Ideally, for a system to work optimally, each speaker should have the exact same way of speaking and have equipment which records the speech to be specific for the user's voice. This instance will not be likely to exist, where we as humans will want the technology to conform to us and not the other way around.

Glossary

All definitions found here are compilations from all of the sources we have used throughout this project.

An **acoustic model** describes the probabilistic behaviour of the encoding of the linguistic information in a speech signal. LVCSR systems use acoustic units corresponding to phones or phones in context. The most predominant approach uses hidden Markov models (HMM) to represent context dependent phones, or speech sounds.

Acoustic phonetics is the study of the physical properties of the sound waves produced by speaking. This involves the measuring and analysis of the speech sound waves.

Articulatory phonetics concerns the study of the physiological mechanisms of speech production.

Automatic Speech Recognition (**ASR**) is the conversion or transcription of speech into the corresponding sequence of words as text. Also known as SR (speech recognition). Speech-to-text (STT) is a derivative of ASR (see *Speech-to-text*).

A **Hidden Markov Model** (**HMM**) is a stochastic (non-predictable) process that satisfies the Markov Property (memory-less). It has a hidden state and an observable state. The hidden state is not observable and is the information that you wish to predict. The observable part is the output of the model. The hidden states are the actual words that were spoken, with the observable state being the acoustic signals.

A **dictionary**, or word lexicon, in the context of speech recognition, is a database of the phones and phonemes the system uses to transcribe the given speech.

A **Language model** models the likelihood of words based on the previous word or words. It is a file that captures the regularities in the spoken language and is used to estimate the probability of word sequences. One of the most popular method is the so called n-gram model, which attempts to capture the syntactic and semantic constraints of the language by estimating the frequencies of sequences of n words.

LVCSR , or **Large Vocabulary Continuous Speech Recognition**, refers to a speech recognition system that is able to recognise many thousands of words. Such systems are able to recognise a continuous flow of speech (or conversational speech).

Markov property - A system has the Markov property if the current state is influenced only by the previous state. Any state preceeding that state has no influence.

Morphemes are the smallest units of language that carry information about meaning or function. These are sometimes referred to as the roots of words, for example, the morphemes book and s form the word books. The extra s carries its own meaning in that it tells us that there is more than one book.

Non-speech – There is a set of all possible sounds that are producible by humans, containing all of the possible speech sounds of which portions are found in the inventory of any human language. Non-speech is the portion of all possible sounds humans can produce that do not occur in the set of sounds found in those known languages. Such sounds include the sound made by inhaling through a corner of the mouth, or the 'raspberry' made by blowing hard on the tongue if you stick it out of your mouth.

A **phone**, also called a speech sound, is any sound used in human language, normally denoted by the phonetic transcription in brackets [ ], such as [e], [ð], or [i].

Phonemes , usually placed between slashes // for transcription, are the phonetic 'building blocks' of morphemes or words, into which non-contrasting phones (speech sounds that, if interchanged, would still have the same meaning) are grouped.

Segments of speech refer to the individual speech sounds, or phones.

Speaker-dependent SR systems are those that are designed for a particular user, and can be trained to better recognise the words they say. Many speaker-dependent applications are also LVCSR.

A **speaker-independent** system is able to handle the speech from many different varieties of speakersit is designed for the average user. Many applications like these have small vocabularies and are not especially designed for continuous speech, rather, they provide a small library of commands and are able to recognise predetermined sentences or statements/commands. Though, some speaker-independent applications are designed for LVCSR, this is usually much more difficult to achieve the same accuracy that may be achieved by a speaker-dependent system.

Speech corpus is a database of speech audio files and text transcriptions used to create acoustic models in speech recognition.

Speech Recognition (**SR**), see Automatic speech recognition.

Speech-to-text (**SST**), is the conversion of speech into its textual representation.

State is defined as a unique configuration of information in a program or machine.

A **Stochastic process** is a collection of random variables representing the evolution of some system of random values over time. Even when the starting variable is known, then there is not a single end variable. The system can evolve in infinitely many ways, or at least several.

Syllables , as you may already know, are units of speech sounds that make up a word, usually consistin of a vowel with a consonant (or set of consonants) on either end. These also sometimes referred to as the 'building blocks' of words. Examples include *run-* and *-ing* in 'running', *rhy-* and *-thm* in 'rhythm', and even whole words such as 'sing' which is considered to be one syllable.

The **velum**, located at the roof of the mouth, is the soft extension of the hard palate, movable by the surrounding muscles. The velum can be lowered or raised to create an opening between the nasal and oral cavities, or close them of from each other.

Bibliography

[1] D. Adams and N. Gaiman. *The Ultimate Hitchhikier's Guide to the Galaxy*. The Random House Publishing Group, New York, NY, USA, 2002.

[2] W. A. Ainsworth and W. A. Ainsworth. *Mechanisms of speech recognition*, volume 85 of *International series in natural philosophy and Pergamon international library*. Pergamon, 1976.

[3] International Phonetic Association. https://www.langsci.ucl.ac.uk/ipa/index.html
The International Phonetic Alphabet chart retrieved from:
http://www.langsci.ucl.ac.uk/ipa/ipachart.html,.

[4] M. J. Ball and J. Rahily. *Phonetics: The Science of Speech*. Arnold and Oxford University Press Inc., London, GB, 1999.

[5] Jacob Benesty, M. Mohan Sondhi, and Yiteng Arden Huang, editors. *Springer Handbook of Speech Processing*. Springer Berlin Heidelberg, 2008.

[6] Paul Boersma and David Weenink. Praat. http://www.praat.org.

[7] Irael Cohen, Yiyeng Huang, Jingdong Chen, and Jacob Benesty. *Noise Reduction in Speech Recognition*, volume 2 of *Springer Topics in Signal Processing*. Springer Berlin Heidelberg, 2009.

[8] Michael H. Cohen, James P. Giangola, and Jennifer Balogh. *Voice User Interface Design*, page Chapters 1 and 10. Addison-Wesley, USA, 2004.

[9] Stephen Cook. Speech Recognition HOWTO, 2002.
Retireved from tldp.org:
http://www.tldp.org/HOWTO/Speech-Recognition-HOWTO/index.html.

[10] G.E. Dahl, Dong Yu, Li Deng, and A. Acero. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, Jan 2012.

[11] David H. Daines. *An Architecture for Scalable, Universal Speech Recognition*. PhD thesis, School of Computer Science, Carnegie Mellon University, USA, 2011.

[12] Mark Gales and Steven Young. Application of Hidden Markov Models in Speech Recognition. *Foundations and Trendds in Signal Processing*, 1(3):195–304, 2007.

[13] Biomedizinische NMR Forschungs GmbH. Real-time MRI - Speaking (English), September 2011. Retrieved from:
http://commons.wikimedia.org/wiki/File:Real-time_MRI_-_Speaking_%28English%29.ogv.

[14] Ed Grabianowski. How Speech Recognition Works, 2006.
Retrieved from HowStuffWorks.com:
http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition.htm.

[15] R. E. Gruhn, W. Minker, and S. Nakamura. Automatic Speech Recognition. In *Statistical Pronunciation Modeling for Non-Native Speech Processing*, pages 5–17. Springer, Berlin, 2011.

[16] J. Heller and C. Buckley. *Catch-22*. Perfection Learning Corporation, 2011. 50th year edition.

[17] Xuedong Huang and Li Deng. *An Overview of Modern Speech Recognition*, pages 339–66. Taylor and Francis Group, 2010.

[18] Docsoft Inc. *What is Automatic Speech Recognition?*, 2009.
Retireved from Docsoft.com:
http://www.docsoft.com/resources/Studies/Whitepapers/whitepaper-ASR.pdf.

[19] B. H. Juang and L. Rabiner. *Speech Recognition—A Brief History of the Technology Development*. In *Elsevier Encyclopedia of Language and Linguistics*. Second edition edition, 2005.

[20] T. Ma, S. Srinivasan, G. Lazarou, and J. Picone. Continuous Speech Recognition Using Linear Dynamic Models. *International Journal of Speech Technology*, 17(1):11–16, 2013.

[21] R McMillan. *How Google Retooled Android With Help From Your Brain*, February 2013.

[22] T. Nilsson. *Speech Recognition Software and Vidispine*, 2013. Master's thesis. Department of Computing Science, Umea University. Sweden.

[23] William O'Grady, Michael Dobrovolsky, and Francis Katamba. *Contemporary Linguistics: An Introduction*. Addison-Wesley Longman Limited, Edinbrough, Scotland, 1996.

[24] David Pisoni and Robert Remez, editors. *The Handbook of Speech Perception*. Blackwell Handbooks in Linguistics. Wiley, 2008.

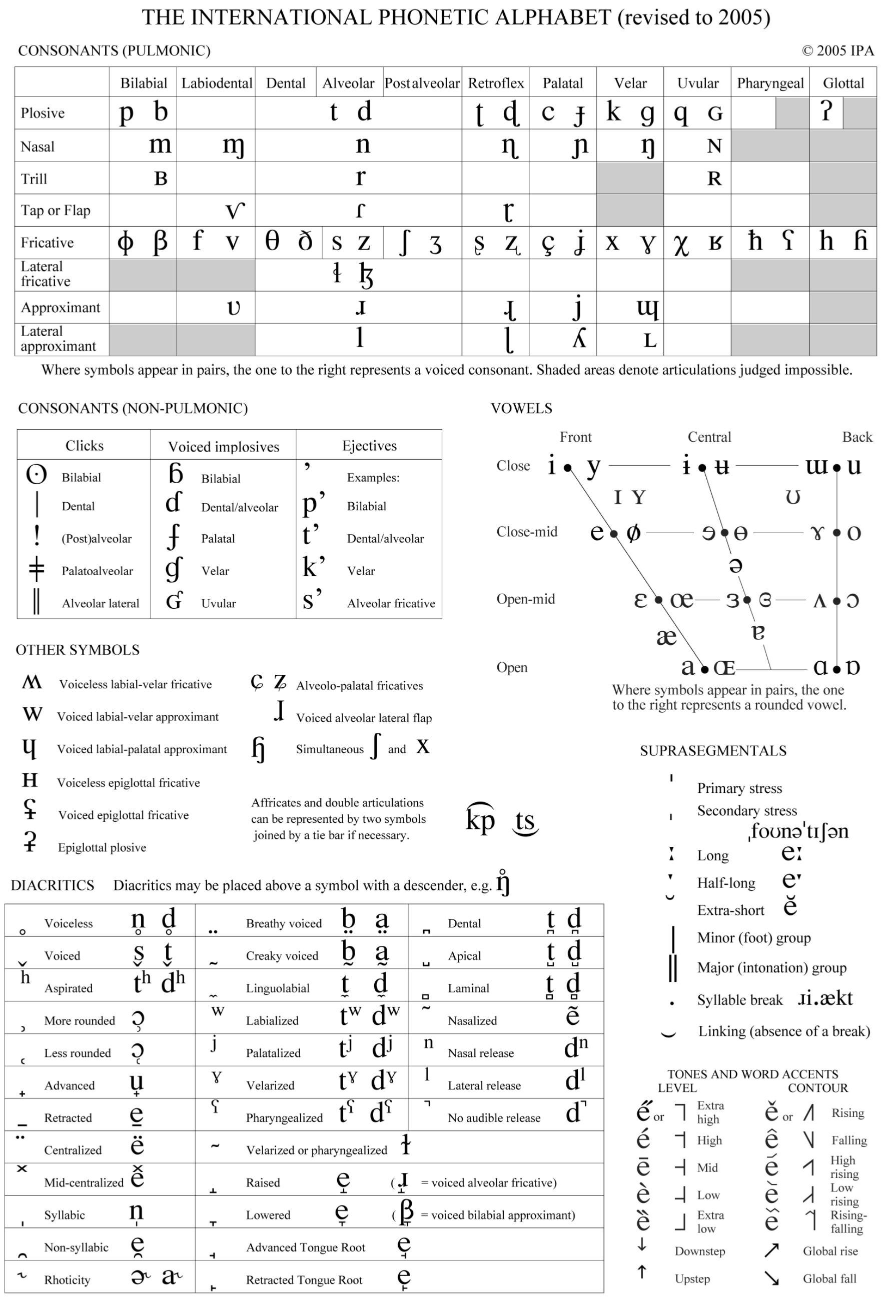
[25] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 7(2):257–286, 1989.

[26] L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, pages 4–15, January 1986.

[27] Javier Ramirez and Juan Manuel Gorriz, editors. *Recent advances in robust speech recognition technology*. Bentham Science Publishers, 2011.

[28] R.C. van Dalen. *Statistical Models for Noise-Robust Speech Recognition*. PhD thesis, Department of Engineering, University of Cambridge, October 2011.

A IPA Chart



THE INTERNATIONAL PHONETIC ALPHABET (revised to 2005)

CONSONANTS (PULMONIC) © 2005 IPA

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Where symbols appear in pairs, the one to the right represents a voiced consonant. Shaded areas denote articulations judged impossible.

CONSONANTS (NON-PULMONIC)

Clicks		Voiced implosives		Ejectives	
ʘ	Bilabial	ɓ	Bilabial	ʼ	Examples:
ǀ	Dental	ɗ	Dental/alveolar	pʼ	Bilabial
ǃ	(Post)alveolar	ʄ	Palatal	tʼ	Dental/alveolar
ǂ	Palatoalveolar	ɠ	Velar	kʼ	Velar
ǁ	Alveolar lateral	ʛ	Uvular	sʼ	Alveolar fricative

VOWELS

Front Central Back

Close i • y ɨ • ʉ ɯ • u

ɪ ʏ ʊ

Close-mid e • ø ɘ • ɵ ɤ • o

ə

Open-mid ɛ • œ ɜ • ɞ ʌ • ɔ

æ ɐ

Open a • ɶ ɑ • ɒ

Where symbols appear in pairs, the one to the right represents a rounded vowel.

OTHER SYMBOLS

ʍ	Voiceless labial-velar fricative
w	Voiced labial-velar approximant
ɥ	Voiced labial-palatal approximant
ʜ	Voiceless epiglottal fricative
ʢ	Voiced epiglottal fricative
ʡ	Epiglottal plosive
ɕ ʑ	Alveolo-palatal fricatives
ɺ	Voiced alveolar lateral flap
ɧ	Simultaneous ʃ and x

Affricates and double articulations can be represented by two symbols joined by a tie bar if necessary. k͡p t͡s

DIACRITICS Diacritics may be placed above a symbol with a descender, e.g. ŋ̊

◌̥	Voiceless	n̥ d̥	◌̤	Breathy voiced	b̤ a̤	◌̪	Dental	t̪ d̪
◌̬	Voiced	s̬ t̬	◌̰	Creaky voiced	b̰ a̰	◌̺	Apical	t̺ d̺
ʰ	Aspirated	tʰ dʰ	◌̼	Linguolabial	t̼ d̼	◌̻	Laminal	t̻ d̻
◌̹	More rounded	ɔ̹	ʷ	Labialized	tʷ dʷ	◌̃	Nasalized	ẽ
◌̜	Less rounded	ɔ̜	ʲ	Palatalized	tʲ dʲ	ⁿ	Nasal release	dⁿ
◌̟	Advanced	u̟	ˠ	Velarized	tˠ dˠ	ˡ	Lateral release	dˡ
◌̠	Retracted	e̠	ˤ	Pharyngealized	tˤ dˤ	◌̚	No audible release	d̚
◌̈	Centralized	ë	◌̴	Velarized or pharyngealized	ɫ			
◌̽	Mid-centralized	e̽	◌̝	Raised	e̝		(ɹ̝ = voiced alveolar fricative)	
◌̩	Syllabic	n̩	◌̞	Lowered	e̞		(β̞ = voiced bilabial approximant)	
◌̯	Non-syllabic	e̯	◌̘	Advanced Tongue Root	e̘			
˞	Rhoticity	ɚ a˞	◌̙	Retracted Tongue Root	e̙			

SUPRASEGMENTALS

ˈ	Primary stress	
ˌ	Secondary stress	ˌfoʊnəˈtɪʃən
ː	Long	eː
ˑ	Half-long	eˑ
◌̆	Extra-short	ĕ
|	Minor (foot) group	
‖	Major (intonation) group	
.	Syllable break	ɹi.ækt
‿	Linking (absence of a break)	

TONES AND WORD ACCENTS

LEVEL			CONTOUR		
e̋ or	˥	Extra high	ě or	˩˥	Rising
é	˦	High	ê	˥˩	Falling
ē	˧	Mid	e᷄	˦˥	High rising
è	˨	Low	e᷅	˩˨	Low rising
ȅ	˩	Extra low	e᷈	˧˦˧	Rising-falling
↓		Downstep	↗		Global rise
↑		Upstep	↘		Global fall

Figure A.1: IPA Chart, http://www.langsci.ucl.ac.uk/ipa/ipachart.html, available under a Creative Commons Attribution-Sharealike 3.0 Unported License. Copyright ©2005 International Phonetic Association.[3]

B Texts for Transcription

Weve chosen some texts to read out into Praat and our speech recognition systems. Below is a listing of all text we have used throughout this report. The longer texts are two short excerpts from the books Catch-22 and The Ultimate Hitchhiker's Guide to the Galaxy.

zip	sip		gnat	bad
zap	sap		not	bade
			note	bead
fat	vat		newt	bed
			knit	bid
cat	hat		neat	bode
pat	bat	mat	net	

Why did she day that?
Why did he say that?
What was that he said?
What was that she said?

Recognise speech.
Wreck a nice beach.
Mechanise peach.

What was that?
Could you repeat that?
That's not what I said.
That is quite the problem.
That was groovy.
I thought that was right.
People don't say that anymore.

'There was only one catch and that was Catch-22, which specified that a concern for one's own safety in the face of dangers that were real and immediate was the process of a rational mind. Orr was crazy and could be grounded. All he had to do was ask; and as soon as he did, he would no longer be crazy and would have to fly more missions. Orr would be crazy to fly more missions and sane if he didn't, but if he was sane he would have to fly them. If he flew them he was crazy and did't have to; but if he didn't want to he was sane and had to. Yossarian was moved very deeply by the absolute simplicity of this clause of Catch-22 and let out a respectful whistle. "That's some catch, that Catch-22," he observed. "It's the best there is," Doc Daneeka agreed.' [16]

'I'm a scientist and I know what constitutes proof. But the reason I call myself by my childhood name is to remind myself that a scientist must also be absolutely like a child. If he sees a thing, he must say that he sees it, whether it was what he thought he was going to see or not. See first, think later, then test. But always see first. Otherwise you will only see what you were expecting. Most scientists forget that' [1]

C Recordings with Praat

A collection of some pictures of the shorter recordings we made with the Praat application. In each figure, the signal is broken up according to how each segment of the words are pronounced, and so we've placed the words spoken at the bottom of each picture, corresponding to where they appear in the signal. The blue lines represent pitch—an increase in slope from left to right corresponds to a raise in pitch.
All recordings were sampled at the same frequency of 44100 Hz.

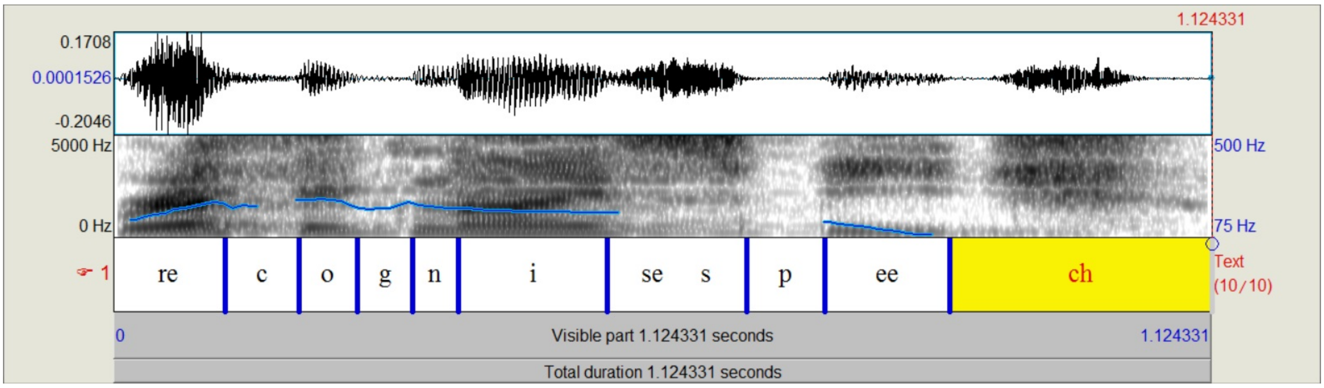


Figure C.1: Bryan's recording of 'recognise speech'.

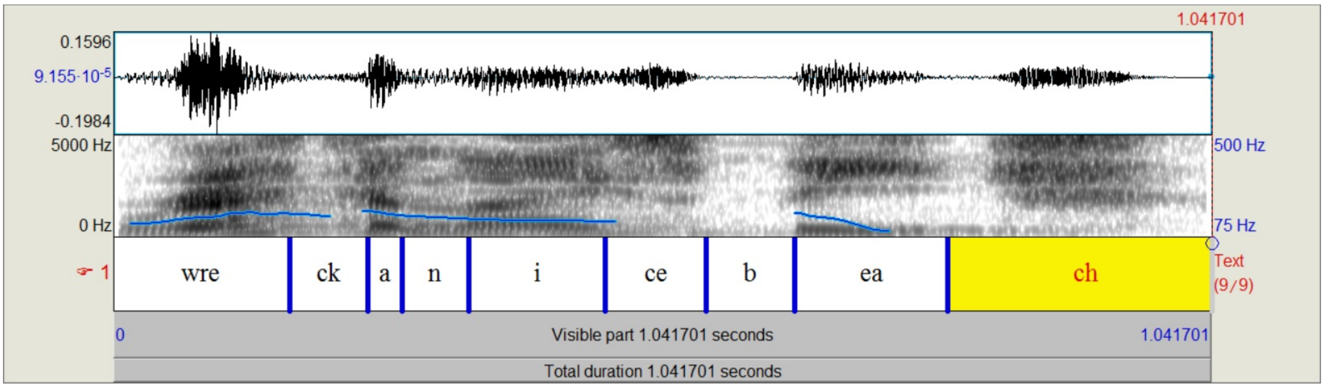


Figure C.2: Bryan's recording of 'wreck a nice beach'.

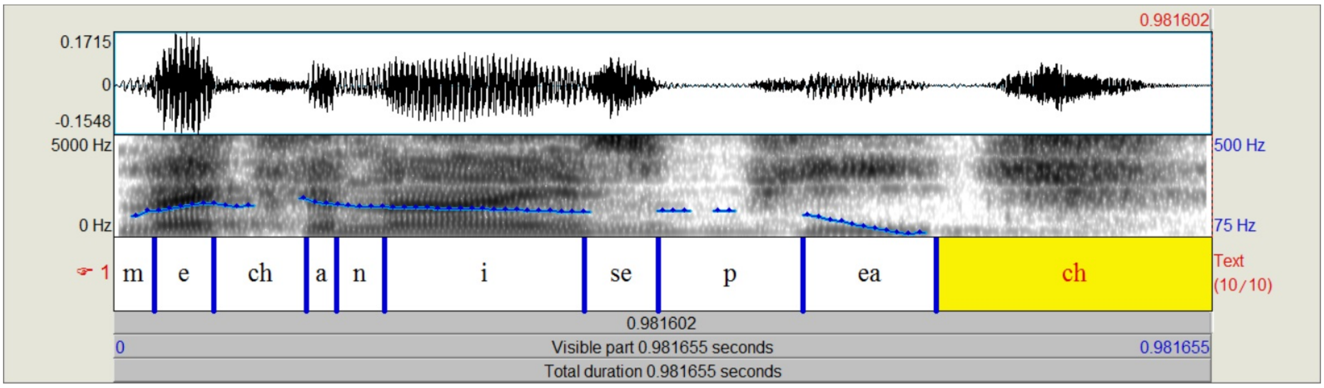


Figure C.3: Bryan's recording of 'mechanise peach'.

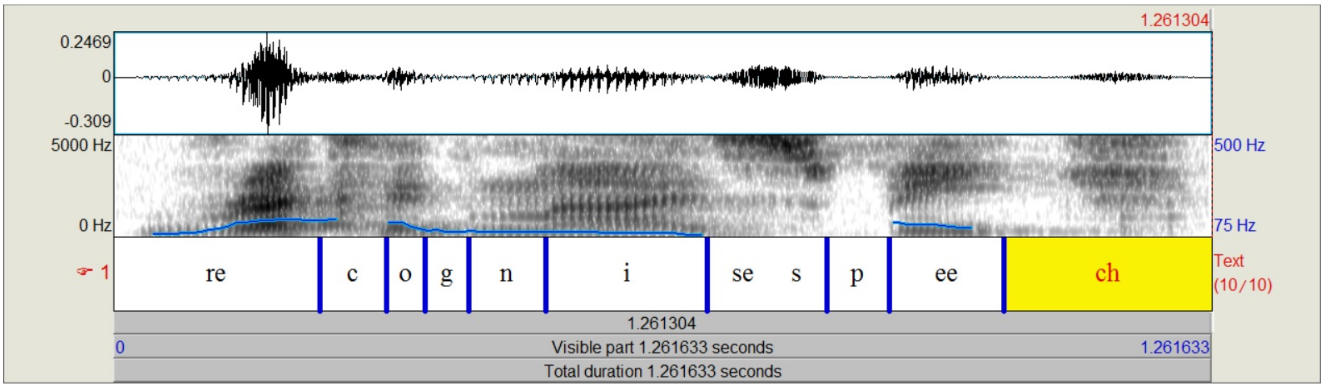


Figure C.4: Sam's recording of 'recognise speech'.

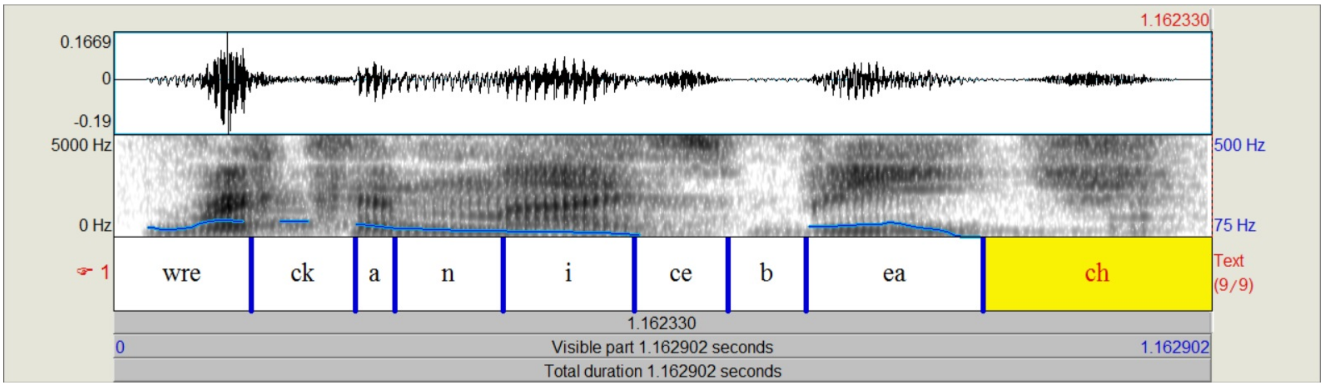


Figure C.5: Sam's recording of 'wreck a nice beach'.

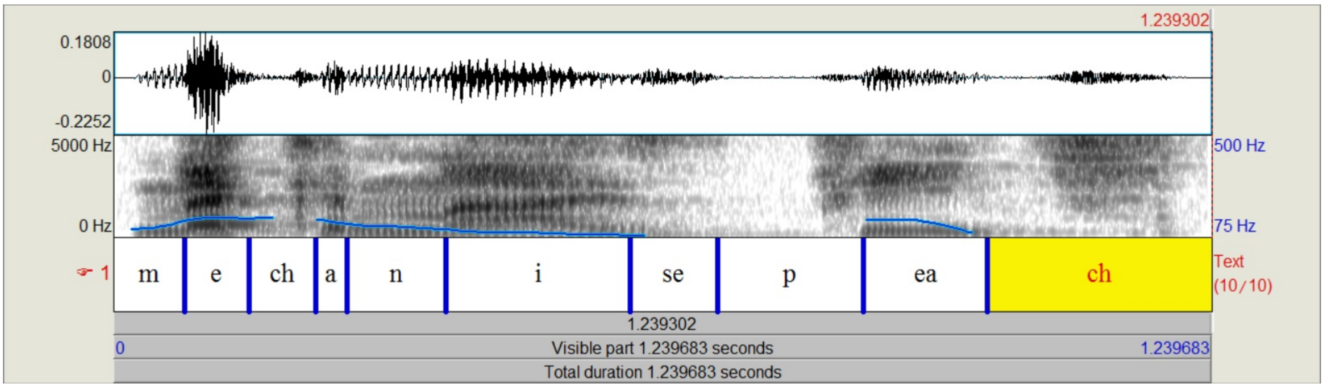


Figure C.6: Sam's recording of 'mechanise peach'.

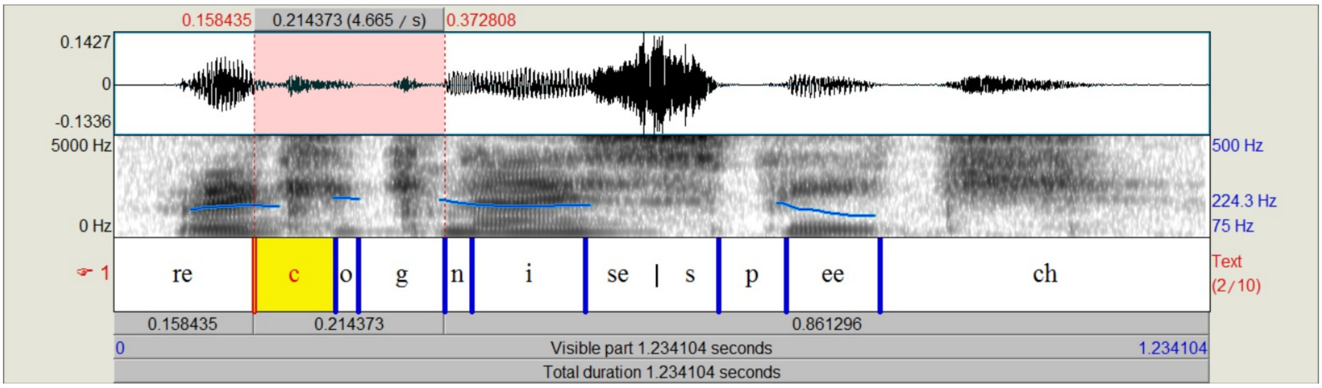


Figure C.7: Savannah's recording of 'recognise speech'.
It was really difficult to figure out where exactly the 'o' is represented within the soundwave. This is because the 'o' here was not really voiced, and so the formant pattern for that is not easily discernable from that of the 'c' and 'g'. Based on the formant patterns and the audio, we've come up with what is seen in the highlighted section of the signal.

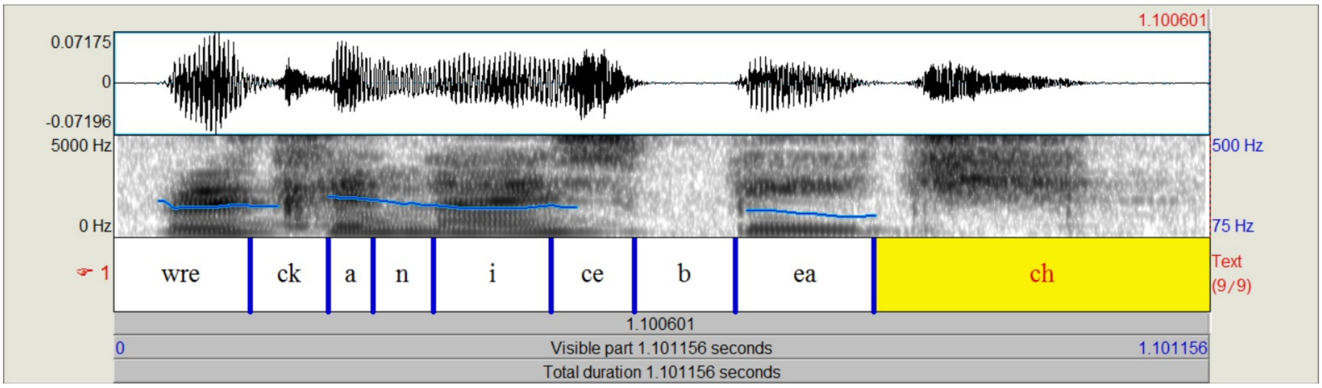


Figure C.8: Savannah's recording of 'wreck a nice beach'.

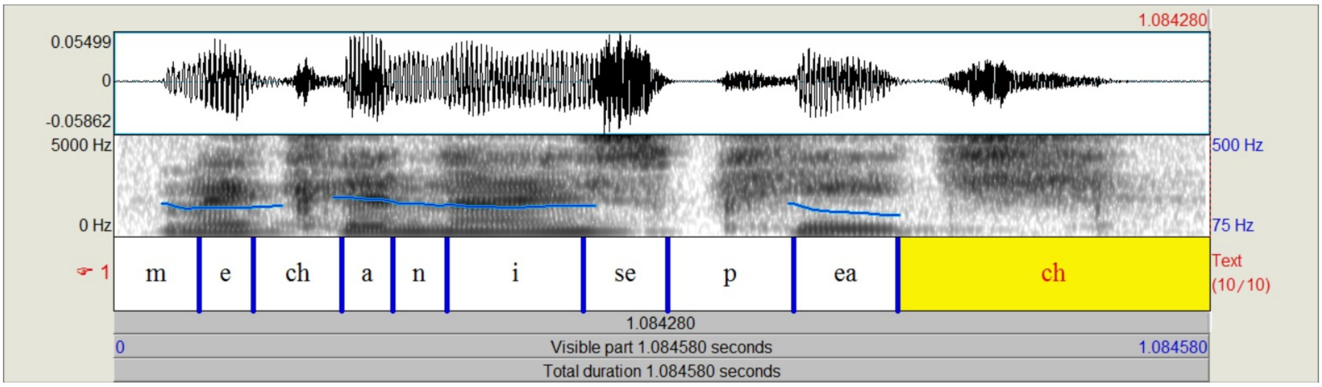


Figure C.9: Savannah's recording of 'mechanise peach'.

D Chrome's Online Dictation Results

The following are the complete transcriptions we received when reading out the words and phrases, as well as the two excerpts from *Catch-22* and *The Ultimate Hitchhikers Guide to the Galaxy*, into the dictation app on Google Chrome. As an extra for the excerpts, we have also read the resulting transcribed text back into the application, giving an interesting coumpounded effect of the errors.

Dictation of the words and phrases by Bryan:

```
Zip, zip .  Zap, sap .  Fat, that .  Cat, cat .  Cats, bats, matt .
Nats not note newt needs .  Nat, not, note, newt, knit, need, net .
Bad, speed, speed, bed, id, food.
```

```
Why did she say that?  Why did he say that?  What was that he said?  What
was that she said?
What was that?  Could you repeat that?  That's not what i said?  This is
quite the problem.  I thought that was right people don't say that anymore.
```

```
Recognise speech.  Recognise speech.  Recognise speech
```

He spoke the phrases more clearly:

```
Why did you say that?  Why did he say that?  What was that he said?  What
was that she said?  What was that?  Could you repeat that?  That's not what
i meant .  That is quite the problem.  I thought that was right.  People
don't say that anymore .  That was ruby.
```

```
Recognise speech .  Wreck a nice beach .  Mechanize peach .
```

The words read by Sam:

```
Zip zip
Zap zap fat matt
Bad bad man
Cat hat
```

```
Nat not note newt meet net newburgh nab.com.au not, note, nude, knit, meath,
.
nap not note newt needs met new era
```

```
Mad.com beeg.com beeg.com bed, vid, bold.  Newburgh
Bad bede bede burn road
Bad bede bede bed being bold
```

Here he spoke the phrases with some prepetition:

```
5 did he say that?  New parrot reagan
Why did you see that?  Newburgh .  Why did she say that ?  What was that
he said ?  What was there she said what was that you said what was that
she said
```

```
Why did he do that?  Wednesday then ?  Christian mark what is there she
said ?  Why did he say that?  Why did she say that?  What is that you said
?  What does he said ?
```

```
what was that can you repeat that ?  That's not what i said ?  That is so
groovy .  I thought that was right .  People say that anymore.

Recognise speech .  Wreck a nice beach .  Mechanised peach .
Recognise beach .  Recognise beach mykonos beach .  Mykonos beach
```

And the same words and phrases as transcribed from Savannah:

```
Zip kama kama zip, .  .
zip, zip
Zip, Sep
Cat, hot you paragraph
Pet, but, met new cars


Net, not, noot noot knit, meat, net new partner fat, speed, speed, bread,
did, food

Why did she say that question mark question mark why did he say that question
mark what was that he said .  What was that she said .  What was that could
you repeat that .  It's not what I said .  That is quite the problem .  I
thought that was right .  People don't say that anymore .

Recognize speech wreck a nice beach .  Make a nice speech .  Recognize speech
.  Wreck a nice beach .  Making a speech .
```

Results from Bryan's reading:

```
That was real and immediate whats the process of a rational mind.  Org was
crazy and could be grand to do was ask, and as soon as he did, he would
no longer be crazy and we have to fly more missions.  Or would be crazy
to fly more missions and saying if you couldn't, but he he was saying he'd
have to fly them.  He was crazy and didn't have to, but if he didn't want
to he was saying and up to.  You said very deeply by the absolute simplicity
of the schools of catch-22 respectful whistle.  That's some cash, that catch
22, he observed.  Is the best there is, dog Daneeka agreed next paragraph

I'm a scientist and I know what constitutes proof but the reason I call
myself by my child who name is to remind myself that a scientist must also
be absolutely like a child.  If he sees a thing, he must say that he sees
it, what date was., think later, dentist pulled off but always see fast.
Otherwise you will only see what you're expecting.  Most scientists to get
that.
```

And a reading by Bryan of the above into the application:

```
Do was ask, and as soon as he did, he would no longer be crazy and we would
have to find more missions.  Let's be crazy 2 final mission and saying if
you couldn't, but he was saying he'd have to fight them.  He was crazy and
didn't have to, but if he didn't want to he was saying and how to.  You
said very deeply by the absolute simplicity of the schools of catch-22 respectful
to.  That some cash.  Catch 22, he observed.  Is the best there is, agreed
next hour ago

And I know what constitutes proof but the reason I call myself by my child
who name is to remind myself that a scientist was also be absolutely like
a child.  If you see something, what beach was., think later, then pulled
```

off bus o'lacy's fast. Otherwise you only see what you're expecting. Most scientists to get that hold up

Sam:

Sex is the face of the angels that were real and immediate whats the process of a rational mind. Or was crazy and could be grounded. Or had to do was ask a semicolon and as soon as he did, he would no longer be crazy and would have to fly from emissions. War would be crazy to fly more missions and seeing if he didn't but but if he was to see through the heavens fly them. If you flew the man he was crazy and didn't have to but if you didn't want to he was saying and had to. Next paragraph new pregg very deeply by the absolute simplicity of this clause of catch-22 and let out a respectful whistle. That's some cash, that catch 22, he observed. It's the best there is, doc Daneeka agreed.

I'm a scientist and I know what constitutes proof. But the reason I call myself by my childhood name is to remind myself that the signs also be absolutely like a child. If you see this thing, he must see that he sees it, what they'd was what he thought he was going to see you're not. See first think later, then test. But always see first. Otherwise you will only see what you're expecting. Most scientists forget that.

And an additional reading by Sam of the result:

Crazy and could be grounded. Or had to do was ask to Nicole and soon as he did come he would no longer be crazy and have to fly from in missions be free to fly more missions each and seeing if he didn't but if he wants to see through the heavens fly them. YouTube the man he was crazy and didn't have to but if you didn't want to but he was seeing and had to. A graph new pregg very deeply by the absolute simplicity of the clothes of catch 22 and let out a respectful whistle. That's some cash that catch 22. It's the best there is doc Daneeka agreed.

I'm a scientist and I know what constitutes proof. But the reason I call myself by my childhood name is to remind myself that the signs also be absolutely like a child. If you see this thing, he must see that he sees it, what the word was what he thought he was going to see you're not. See first think later, then test but OC first. Otherwise you will only see what you're expecting. Most scientists forget that.

Savannah:

There's only one catch and that was catch-22, which best of the Dead Concentra one's own safety in the face of danger said we're really mediate's was the process of the rest of my. Or was crazy into the ground. What have to do is ask and it seems she did, he would no longer be crazy when have to find more missions or would be crazy to find more missions in seeing if you didn't, but if he was seeing you would have to find him. He flew them he was crazy and he didn't have to, but if you didn't 12 she was seen in Hampton. Very deeply buddy Ebsen simplicity of this close to catch 22 in another respectful whistle. That's some cash, that catch 22 come off your. If the best there is, duck Tanika.

What constitutes proof. But the reason call myself on my child his name is to remind myself that a scientist must also be like a child. She sees a thing, you must see that he sees it, whether it is going to see or not think later then test. Otherwise you only see what you were expecting. Was trying to forget that

Savannah's second reading:

When soon safety in the face of danger we really mediates was the process of the rest of my. Or was crazy into the ground. What has to do is ask and seems she says, he knew longer be crazy when have to find more missions or would be crazy to find more missions in CA if you didn't come up with if you were seeing you where to find him crazy and didn't have to, but if you didn't 12 she was empty. Very deeply buddy Ebsen simplicity of this close to catch 22 another. That's the cash, but catch 22 come off your. The best there is, . Newport

What constitutes proof. But the reason of myself on my child name is to remind myself that a scientist must also be like a child. She sees a thing, you must see that he sees it, where it is going to see your Knox and glitter Pinterest. Expecting couple was trying to forget but

E Windows Speech Recognition Results

The following are the transcriptions from dictating the words and phrases and the two excerpts from *Catch-22* and *The Ultimate Hitchhikers Guide to the Galaxy* again. This was the result of reading that text into Notepad, using Windows Speech Recognition (WSR). After receiving the first transcription, we did a reading of that back into WSR.

Dictation results of the words and phrases by Bryan:

```
As in saint
Zzap so
The fact that
And Nats, that's and saint.  Hats, doubts, met.

Nats not know what you next need Nats.  And an Nats, notts, note, Newt,
and it on a neat, Nats,.  Bad, Reid, the, beds, be it, though.

And said, saint.  And slap, slap.

Why did she see that?  Why did he say that?  What was that he said?  What
was that she said?  What was that?  Could you repeat that?  That's not what
I said.  That is quite properly.  I thought that was right.  That was truly.
People don't say that any more.

Recognize speech..  Recognize Beach.  Mechanize P each.
And and Recognize speech.  All thought Mechanize P each.
```

The words and phrases read by Sam:

```
Zero, and soon, and all
All,:  San
France, that's
Pounds, but, Matt
Cats, at

Nets, notts, lincs, Newt, Newt, needs, net
Bound, read, read, then, the I'm a big, bold.

10?  Why did she seem to?
Why did he zero?  Why did he zero?
One is that his own?  What was there isn't?
Was this reason?  What was that research?

Was that?
Any repeat the question mark
None wasn't Korean.  Must not Wallace and.
That is all the.  That is all around me.
I thought I was right..  I thought I was right.
You've seen by any and all.
Also minimal.  You zero been in

Recognized beach.  Recognise means.
Recognise beach.  Recognise beach.
```

And the same words and phrases as transcribed from Savannah:

```
As in, soup, no, sir,
Zip, zip, sir, sir,

Saint, saint.
Step, step.  And can
Cent, that.  Fact, that.
, Had, the care weekend and care
Praat, that Kohl met.

No, not, no, me and, need and the need.  Now and not named and need need
net.  No, not, colour need, nicked, need, not.
Then come the and can be had from they had come by and colour code.  Darren.
Read.  Need.  Good..  Lead.  Road.

Where did she see the?  Where did he see in the?  Was that he's an?  What
was that she's an?  What was up?  Could you repeat that?  Does not wish
to.  Is not wasted.  I thought that was red.  So close range.  People dont
see that any more.  Did not answer him any more.

Recognize speech.  Recognize Beach.  Mykonos kitsch.  Mykonos kitsch.
```

Excerpt dictation by Savannah:

```
There was only one catch and house cats 22, were specified that consensual
and some safety in the face of danger is the real enemy it was suppressed
some rational mind.  Or was crazy into the ground.  All had to do was ask,
and as soon as he did, he would no longer because I have to plan the missions.
All it would be crazy to slimmer missions and see me she didn't, the issue
is seen here is designed to create a friend and he was crazy after, but
if he didn't want to be missing and had two.  ESI in this proves thinking
by the accents and Christians cause of catch 22 and lets out a respect for
whistle.  But some cats, that catch 22, he observed.  It's the best there
is, that many care and tied

I am as scientist and who would constitute proof.  But the reason I call
myself to my children mean is to remind myself that scientists must also
be absolutely like a child.  That he sees a thing, Siemens said he sees
it, whether it was what he thought he was going to see or not.  Seafirst,
simply, and test came early Seafirst.  Otherwise you only see what you were
expecting.  Most scientists forgot that could
```

And a transcription when Savannah read the results back into WSR:

```
It was only one catch as catch 22, was specified that consensual Thompson
see in the face of danger is the real enemy of west coast some rational
mind game or was crazy into the ground in all here to do is ask, and as
soon as he did, he would no longer because I have to plan the nation's paid
Hollywood because it's so than ever since and Siemens CD, issue is seen
here designed to create a friend and he was crazy after, that if he didn't
want to be missing content to.  ESI in the spruce thinking by accents and
Christians cause of catch 22 and let's not a respect for whistle.  But some
cats, that catch 22, he observed.  It's the best there is, that many care
and turned

I am a scientist and he would constitute proof.  But the reason I call myself
```

```
to my children mean is to remind myself that scientists might have missed
also be absolutely like a child.  But he sees a thing, CNN's said he sees
it, where was what he thought he was going to see or not.  Seafirst simply,
contests can earn the Seafirst.  Otherwise you won't see which are expected.
Most sciences from that that could
```

And another reading by Savannah after attempting to train the system to her voice:

```
There is only one catch and I was catch 22, which specified that a concern
for one's own safety in the face of danger still real media was the process
of rational mind.  Or was crazy into the ground it.  All yet to do is ask,
and as soon as he did, he would no longer be crazy or have to find more
missions.  Or would be crazy to fly more missions in saying that if he didn't,
but if he was saying he would have to fly them.  If he flew them he was
crazy and didn't have to, but if he didn't want to hear the same and had
to.  ESI in this move very deeply by the absolute simplicity and his cause
of catch 22 and let out a respectful whistle.  That's some catch, that catch
22, he observed.  It's the best days, Doc Denny Kerr agreed.

I'm a scientist and I know what constitutes proof.  But the reason I call
myself by my childhood name is to remind myself that a scientist was also
be absolutely like a child.  If he sees a thing, he must say that he sees
it, whether it was what he thought he was going to see or not.  Seafirst,
think later, then test.  But always see first.  Otherwise you only see what
you are expecting.  Most scientists forget that.
```

Bryan's reading of the excerpt from The Hitchhiker's Guide to the Galaxy with the relatively trained WSR:

```
Imus scientist and I know what constitutes proof.  of the reason I call
myself I was halted name is to remind myself and scientists must also be
obsolete me like a child.  if he sees the reigning, king city that he sees
that, whether it was what he thought he was going to C1, OC frost think
we can, then test, the woolly C frost.  otherwise you'll only see what you
were expecting.  of scientists forget that three com
```