CENTRE *for* DOCTORAL TRAINING *in*
CYBER SECURITY

CDT Technical Paper

01/16

Security Vulnerabilities in Speech Recognition Systems

Mary Bispham

Security Vulnerabilities in Speech Recognition Systems

Mary K. Bispham, Michael Goldsmith and Steve Moyle
Centre for Doctoral Training in Cyber Security
University of Oxford
Department of Computer Science
mary.bispham@cybersecurity.ox.ac.uk

Abstract. *The aim of this project is to develop a generic methodology for black-box testing of speech recognition systems by seeking to identify homophones or similar sounding words which may be misrecognised by a system as command words. A series of experiments are performed with two different speech recognition systems. The data used in the experiments are a set of words from the vocabulary of a controlled natural language, Attempto Controlled English, together with a set of words not permitted in that controlled language. Several instances are identified where a 'legal' Attempto word is misrecognised as an 'illegal' word. This demonstrates the feasibility of an attack whereby an attacker might seek to find apparently innocuous words which are misrecognised by a speech-controlled system as commands, thus enabling the attacker covertly to prompt the system to perform an unauthorised action. In such a situation, the attacker would identify a set of command or 'target' words used to control a system (the equivalent of the 'illegal' words in the experiments with Attempto) and would seek to find a set of 'adversarial' words which are misrecognised by the system as a target word. In a real-life attack, an attacker might seek to find words which are misrecognised as command words for a digital assistant such as Siri or Cortana, or as command words for a voice-controlled device in the Internet of Things.*

Keywords: automatic speech recognition, human-computer interaction, Internet of Things, cyber security

1 INTRODUCTION

The use of speech recognition as a modality of human-computer interaction is becoming more widespread. Google have sought to develop a capability for web search by voice (see Schalkwyk et al. (2010)). Digital assistants such as Apple's Siri or Microsoft's Cortana allow users to perform various tasks by voice command, such as dictation, playing music, or making a phone call. Start-up company Wit.ai is aiming to provide voice control capabilities for devices in the Internet of Things such as kitchen appliances[1]. Applications of speech recognition are also being extended to control of life-critical systems, including drones, military robots, and surgical instruments (see Craparo and Feron (2004), Barber et al. (2014), Nathan et al. (2006)). The potential benefits of speech control were dramatically demonstrated in a recent incident in which a mother ordered her smart-phone to call emergency services by voice command whilst performing emergency

[1] https://www.technologyreview.com/s/531936/voice-recognition-for-the-internet-of-things/

first aid on a child[2]. Lison and Meena (2014) claim that the potential of speech recognition is currently 'greatly underexploited'; thus it can be expected that development of new speech-controlled systems will continue to be pursued in future.

Whilst the potential benefits of speech recognition technology are not in doubt, the application of such technology in new contexts brings with it new security considerations requiring to be addressed. Allowing a computer system to be accessed by speech command represents a significant increase in attack surface. Some security measures have been implemented in speech recognition systems, such as 'repeat back' requirements to confirm the user's intention, and implementation of individual speaker recognition intended to ensure that the system will respond only to the voice of a specific individual. However, 'repeat back' may not be an option where speech recognition is applied in a real-time context, and speaker recognition is likely to be impracticable in many instances where the system is required to be usable by different people. The potential vulnerabilities associated with speech controlled systems been highlighted by Dhanjani (2015), who describes a security vulnerability identified in Windows Vista which allowed an attacker to delete files on a victim's computer by playing an audio file hosted on a malicious website or sent to the victim as an email attachment. Dhanjani speculates that the potential for such attacks is magnified with the increasing use of speech recognition technology in the Internet of Things. He postulates a hypothetical attack on Amazon's Echo, a device designed to be used for voice control of home appliances via digital assistant 'Alexa'. This hypothetical attack involves a piece of malware consisting of JavaScript code which plays an audio file giving a command to Alexa if there has been no user activity on the mouse or keyboard after a certain period of time (thus aiming to play the file at a time when the user may be away from their computer and therefore will not hear the audio command being played).

The feasibility of the hypothetical attack foreseen by Dhanjani was demonstrated in a subsequent real-life incident, where examples of commands which might be given to Alexa were spoken in a radio broadcast about Amazon Echo, having the unintended effect of issuing commands to the Echo devices of users listening to the broadcast in their homes, for example issuing commands to adjust thermostats[3]. Whilst the remote take-over of the Alexa functionality in individual homes was unintended in this incident, a malicious attacker might just as easily gain access to a voice-controlled system in the same way.

The hypothetical attack described above does not rely on any errors or bugs in the speech recognition system; it simply exploits the intended functionality of the system for an unintended purpose. There are also potential attacks associated with the proneness of speech recognition systems to make errors. Cutajar et al. (2012) point to the significant challenges of achieving accurate speech recognition, including background noise, difficulty in identifying word boundaries, and variability of individual speech characteristics such as accent. To this list might be added the presence of homophones and similar-sounding words within natural language. It cannot be denied that notwithstanding such difficulties the performance of automatic speech recognition has shown continuous improvement. In 2002, researchers at IBM Watson Research Centre reported a word error rate of 32.7 percent on a voice-mail transcription task (Padmanabhan and Picheny (2002)), whereas by 2016, the same Centre was reporting a word

[2] https://nakedsecurity.sophos.com/2016/06/08/mother-saves-1-year-olds-life-while-iphones-siri-calls-for-an-ambulance/

[3] http://fortune.com/2016/03/12/npr-amazon-echo/

error rate of 6.9% on the Switchboard corpus of transcribed telephone speech[4]. However, it is possible that a certain degree of inaccuracy is inevitable, and that improvement in speech recognition accuracy beyond a certain point will prove elusive. The developers of Sphinx, an open source software library for speech recognition, have stated that automatic speech recognition always falls short of absolute accuracy.[5] It is worth noting that the same is true of human speech recognition. The paper by Padmanabhan and Picheny notes that the human word error rate in recognition of conversational telephone speech is estimated at around 5.0 percent.

The persistent inaccuracy of automatic speech recognition may represent an attack vector for an actor seeking to gain access to a speech-controlled system for malicious ends. It is possible to imagine, for example, an auditory equivalent of 'typo-squatting'. 'Typo-squatting' is the criminal practice of hosting malicious websites which have the same name as a legitimate website barring a small spelling error, the aim being to lead a certain number of users seeking to visit the legitimate website to the malicious site instead. The victim might then be prompted to enter personal details on the malicious site, or be exposed to attack via a 'drive-by download' of malware. The auditory equivalent of 'typo-squatting' would be to host a malicious website with a same-sounding or similar-sounding name as a malicious website. This might lead users browsing the web by voice command to unintentionally visit the malicious website if the speech recognition system being used for browsing misrecognises the name of a legitimate website as the name of a malicious one. Alternatively, an attacker might seek to identify words which might be misrecognised by a speech-controlled system as a command. Discussing the security vulnerability identified in Windows Vista described above, Dhanjani claims that this vulnerability might not have been considered serious on the basis that any audio attack on a system would be heard and thus immediately identified by a user. However, an attacker using words apparently not addressed to the speech-controlled system would be able covertly to access the system without drawing the victim's attention to their activities. Such attacks on a system by voice command would be both untraceable and unattributable.

2 BACKGROUND

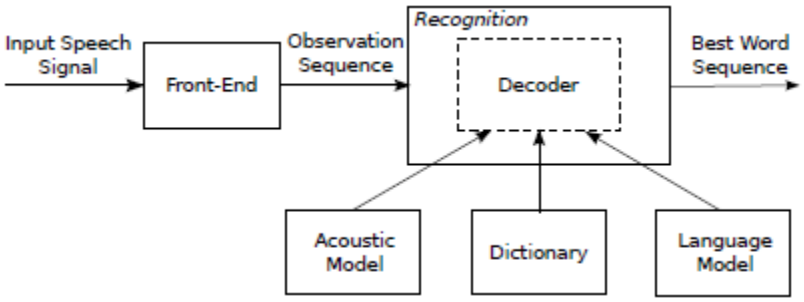
The process of automatic speech recognition begins with sampling of a speech sound wave at regular short intervals (typically 25 milliseconds). A spectrum (a plot of frequency and amplitude values) for each sample is stored in digitised form, and a set of features characteristic of different speech sound units (phones) is then extracted from each digitised sample. The most common feature extraction method is the Mel Frequency Cepstral Coefficients (MFCC) method, which represents each sample spectrum as a 39-dimensional vector (the mel scale being a sequence of frequencies in which all steps in the scale are perceived by humans to be equally far apart). These features are then provided as input to the 'decoder', which generally consists of three components, namely an acoustic model, a phonetic dictionary and a language model (see Figure 1). The acoustic model typically uses a machine learning classification method to

[4] https://developer.ibm.com/watson/blog/2016/04/28/recent-advances-in-conversational-speech-recognition-2/

[5] "All modern descriptions of speech are to some degree probabilistic. That means that there are no certain boundaries between units, or between words. Speech to text translation and other applications of speech are never 100% correct. That idea is rather unusual for software developers, who usually work with deterministic systems. And it creates a lot of issues specific only to speech technology." (from http://cmusphinx.sourceforge.net/wiki/tutorialconcepts)

determine the phone (or larger unit of speech such as a syllable) which is likely to have generated the spectral information found in a given speech segment. The most commonly used classification method is the Hidden Markov Model (HMM), with a Gaussian Mixture Model (GMM) being used to calculate the likelihoods associated with each HMM state. Artificial Neural Networks (ANNs) are also increasingly used as an alternative or complementary method (see Hinton et al. (2012)). The second component of a speech recognition system, the phonetic dictionary, maps the phonetic sequences identified by the acoustic model to a natural language vocabulary. Although the size of the vocabulary which a speech recognition system is able to recognise may be very large, it will generally be a 'closed' vocabulary system, i.e. the system will not have the capability to recognise words which are not included in its phonetic dictionary, and may instead misrecognise such words as words which are present in its dictionary, or else tag them as unknown. Qin (2013) has explored possibilities for a 'self-learning' speech recognition system which would be able to incorporate new words in its dictionary as it encounters them in interactions with the world. The third component of a speech recognition system, the language model, assesses the probability of the presence of an individual word in a specific speech context. The language model is typically based on n-gram probabilities extracted from a large corpus of training data.

Figure 1: *Architecture of an automatic speech recognition system (from Qin (2013))*



Some prior work has sought to address possible security vulnerabilities associated with speech recognition systems. Papernot et al. (2016) have demonstrated a black-box attack methodology against machine learning classification systems. Whilst their work is performed in the context of image classification, their results are also relevant to speech recognition systems based on the same machine learning methods. The authors show that it is possible to lead a machine learning classifier to misclassify images by making very small modifications to the image which would be imperceptible to a human. This is done by collecting from the targeted system the set of outputs for a given set of inputs, and then using these outputs to generate a set of 'adversarial samples' with a substitute classifier, through application of a dataset augmentation method and an algorithm for calculating small perturbations to input which is correctly recognised by the system. The authors use the example of traffic sign recognition by self-driving cars to illustrate the value of their work, stating that small modification of a 'Stop' sign not noticeable to a human driver might lead to the 'Stop' sign being misrecognised by the automated vehicle. This is comparable to misrecognition by a speech recognition system of a command word used to control a physical system. Carlini et al. (2016) also present the results of a black-box experiment, in which it was possible to achieve recognition by Google's speech recognition of recorded commands such as 'Ok Google' which had been distorted by an 'audio mangler' so as to be perceived by humans as mere noise.

Some research has focussed on the linguistic input to speech recognition systems. One example is the development of an artificial language named ROILA for human-robot communication (see Mubin et al. (2010) and Mubin et al. (2014)). The developers of ROILA used a genetic algorithm to generate a vocabulary of acoustically distinct words for the artificial language, seeking thus to avoid issues of phonetic ambiguity as found in natural language. Kaljurand and Alumäe (2012)) describe the development of an architecture for mobile phone apps which combine controlled natural language with speech recognition capabilities. They point to the additional challenges in using controlled natural language in a speech-based application as opposed to a text-based application, noting the issue of homophones which can be distinguished in written but not in spoken language, such as in the example of 'Pii', which is an Estonian place-name but may by misrecognised as the number π by a calculator in the mobile phone. The authors state that measures need to be put in place to ensure that such instances of ambiguity in spoken language are eliminated from a controlled natural language which is to be used in speech recognition systems. In the context of voice control of home devices in the Internet of Things, Mehrabani et al. (2015) suggest the use of domain-specific personalised language models to avoid misrecognition errors which may result from use of a generic language model based on statistical word probabilities in unrelated domains. The authors provide some examples of commands which may be misrecognised by a generic model, claiming that the words 'gym' and 'dim' in the commands 'Turn gym light on' and 'Dim dining room light' might be misrecognised as 'Jim' by a generic system. The authors argue that such errors can be avoided by implementation of a more restricted language model which only recognizes words which are relevant to a specific home context.

3 METHODOLOGY

The experimental structure for this project sought to identify instances of misrecognition of a potential 'adversarial' word as one of a short set of 'target' words. The approach represents an application in a linguistic context of a computer science testing technique known as 'fuzzing' (see for example Oehlert (2005)). The experimental dimensions were two sets of potential 'adversarial' words of different sizes, two speech synthesis systems, and two speech recognition systems. The experimental set-up consisted of a chain, whereby a word list not containing any 'target' words (i.e. a set of potential 'adversarial' words) was provided as input to a speech synthesis system, the synthesized speech sound from each word was subsequently provided as input to a speech recognition system, and the resulting reverse transcription of the original input was then compared to the original input to find any instances of misrecognition of a speech sound as one of the 'target' words (see Figure 2). This experimental structure was implemented as a Python program, as detailed in Section 5.

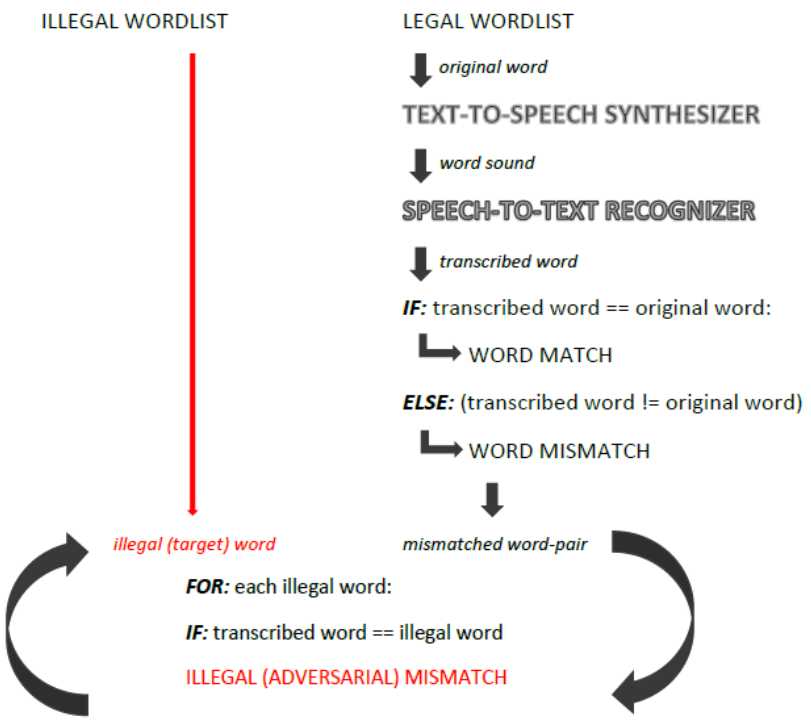
Five experiments were performed, using two speech synthesis systems and two speech recognition systems. In the first four experiments, the different synthesis and recognition systems were combined pairwise and tested with the same set of potential 'adversarial' words. The last experiment tested one of the system combinations used in the previous experiments with a different, larger, set of potential 'adversarial' words. The list of 'target' words was the same for all five experiments. It needs to be noted that in the context of these experiments, the set of potential 'adversarial' words are also referred to as 'legal' words, in the sense of being permitted in the Attempto controlled natural language, and the set of 'target' words as are also referred to as 'illegal' words, in the sense of being forbidden in the Attempto controlled natural language (see Section 4 for further details of the Attempto data used in the experiments).

The output of the experimental process consisted of a list of word matches, a list of word mismatches, and a list of word pairs extracted from the list of word mismatches which represented conversion from a 'legal' to an 'illegal' i.e. 'target' word (the 'legal' words which were misrecognised as an 'illegal' word thus representing a set of 'adversarial' words.). A phoneme string was generated for each word identified as 'adversarial' and for the corresponding 'target' word.

For each mismatched word pair representing a legal-illegal word conversion, the Levenshtein distance and ratio for their orthographic strings were calculated. The Levenshtein distance is the minimum edit distance between two strings (whereby insertion and deletion represent a score of 1 and substitution represents a score of 2). The Levenshtein ratio is the combined length of the two strings being compared minus the Levenshtein distance between them, divided by the combined length. Thus the Levenshtein ratio represents a measure of the edit distance between the two strings relative to their length. The lowest possible value for the ratio is 0, indicating a maximum edit distance where the edit distance is equivalent to the combined length of the strings. The highest possible value for the ratio is 1, indicating the minimum edit distance of 0, i.e. two identical strings. It was recognised that the Levenshtein distance and ratio between orthographic strings does not represent a reliable measure of the difference between two words in terms of their speech sound. More sophisticated measures of phonetic distance, based on differences between the features of articulation of a speech sound as performed by a human, are discussed in Pucher et al. (2007)).

For the last experiment with a large list of controlled vocabulary words, a statistical test was performed to determine the probability of the observed number of legal-illegal word misrecognitions. This involved calculating an expected value for the legal-illegal word pairs based on the binomial distribution, and calculating a probability for the actual value by approximation of the binomial distribution to the normal distribution.

Figure 2: *Experimental set-up (schayn.py)*



4 DATA

The experiments performed in this project used as input a set of words from the vocabulary of Attempto Controlled English, i.e. 'legal' Attempto words, and a set of words not permitted in Attempto, i.e. 'illegal' words. Attempto is a restricted subset of English (see Fuchs and Schwitter (1996)). This controlled language is designed to support the creation of sentences which, although written in natural English, can be translated into an unambiguous logical form underlying the natural language text. Attempto sentences can also be translated into executable form in the Prolog programming language. Some words are 'illegal' in Attempto because they do not fit with the logic of the controlled language. An example of the restrictions imposed by Attempto is that the language only allows creation of sentences written in the third person and in the present tense. Consequently first person pronouns such as 'I' and 'me' are not permitted in Attempto. Also, in accordance with the requirement that Attempto sentences should be capable of being represented in first-order predicate logic, only existential and universal 'quantifiers' are permitted. Thus the words 'somebody' and 'something' (represented by logical symbol ∃) and the words 'everybody' and everything (represented by logical symbol ∀) are permitted, whereas the words 'anybody' and 'anything' are not. The full list of words declared 'illegal' in Attempto is the following set of 15 words:

any, anybody, anything, this, these, I, me, my, mine, yours, we, us, our, ours, theirs

It was noticed that the two second person pronouns 'you' and 'your' are not declared 'illegal' in Attempto. It was assumed that the reason for this is that these two pronouns can also be used in the impersonal sense of 'one' and 'ones', as well as as second person pronouns.

The first four experiments used as input the set of 15 words forbidden in Attempto Controlled English, and a set of 1,214 words from the Attempto vocabulary space, the latter incorporating a set of words specified as 'function words' in Attempto as well as the full vocabulary of a small in-built lexicon distributed as part of the Attempto Parser Engine. The fifth experiment used as input the same set of 15 'illegal' words, and a set of 77,689 words from the Attempto vocabulary space, the latter incorporating the Attempto 'function words' referred to above as well as the full vocabulary of a large lexicon distributed for Attempto separately from the Attempto Parser Engine. (NB 'Words' as referred to here includes all individual Attempto lexicon entries, which may consist of more than one individual word, eg. 'relate to'. See Appendix C for further details of the word lists.)

5 SOFTWARE

Two speech synthesis systems were used. The first was the open source eSpeak software for speech synthesis. eSpeak includes a functionality to output a phoneme string for a given segment of synthesized speech (see Appendix D for details of the phoneme symbols used by eSpeak). The synthesized 'voices' produced by eSpeak system are generated by formant synthesis, a 'formant' being a visible 'peak' on a spectrogram which indicates high amplitude for a particular frequency in the speech signal at a given point in time. eSpeak also facilitates use of synthesized voices developed in the MBROLA project[6], whereby text to phoneme translation for use by these 'voices' is provided by eSpeak. MBROLA voices are based on diphone synthesis, which uses databases of actual speech and concatenates segments of speech representing

[6] http://tcts.fpms.ac.be/synthesis/mbrola.html

transitions between sounds. For these experiments, MBROLA voice us-1, representing a female US English speaker, was used with eSpeak. A Python wrapper for eSpeak was used. The second speech synthesis system used in the experiment was IBM Watson Text-to-Speech. This system is based on concatenation of segments of human speech[7]. IBM Watson Text-to-Speech provides a variety of different synthesized voices. For these experiments, the 'Allison' voice was used, representing a female US English speaker. IBM Watson Text-to-Speech is a cloud-based system which is made available to developers in the Watson Developer Cloud.

Two speech recognition systems were used. The first was Pocketsphinx, a small-memory version of the open source Sphinx software for speech recognition which has been made available by Carnegie Mellon University. Sphinx is an HMM-based speech recognition system which includes a default language model and phonetic dictionary (CMUdict) for US English. For this experiment, a Python-wrapped version of Pocketsphinx was used. The second speech recognition system used in the experiment was IBM Watson Speech-to-Text, which, like IBM Watson Text-to-Speech, has been made available to developers on the Watson Developer Cloud (see Appendix B for details of all third party software used).

A Python program for linking text-to-speech and speech-to-text functionality (schayn.py) was written and adapted for use in the different experiments (see Appendix A for details of the source code files). The Python code included functionality for outputting eSpeak phoneme strings for original and transcribed words, as well as for calculating the Levenshtein distance and ratios between mismatched original and transcribed words. eSpeak phoneme strings were outputted both for the experiments with eSpeak and for the experiments with IBM Watson Text-to-Speech, as IBM Watson Text-to-Speech does not include a facility for outputting phoneme strings.

6 RESULTS

Instances of illegal-legal word conversion were found across the various combinations of speech synthesis and speech recognition technologies. The overall word error rates were very high, although there were notable differences between error rates in the different experiments. In the eSpeak to Sphinx experiment, the number of illegal-legal word conversions identified was 3, and the overall word error rate was 63.7%. The illegal-legal word conversions and the corresponding Levenshtein distances and ratios were as follows:

original	*transcribed-illegal*	*phonemes-original*	*phonemes-transcribed*	*Levenshtein-orthographic*	*orthographic-ratio*
boss	*us*	*b'Os*	*'Vs*	*4*	*0.333*
exactly	*we*	*Egz'aktli*	*w'i:*	*7*	*0.222*
years	*yours*	*j'i@3z*	*j'o@z*	*4*	*0.600*

In the eSpeak to IBM Watson Speech-to-Text experiment, the number of illegal-legal word conversions identified was 4, and the overall word error rate was 53.0%. The illegal-legal word conversions and the corresponding Levenshtein distances and ratios were as follows:

[7] https://developer.ibm.com/watson/blog/2015/02/09/ibm-watson-now-brings-cognitive-speech-capabilities-developers/

original	transcribed-illegal	phonemes-original	phonemes-transcribed	Levenshtein-orthographic	orthographic-ratio
eye	I	'aI	_#X1'aI	4	0
hour	our	'aU@	'aU3	1	0.857
or	our	'O@	'aU3	1	0.800
he	we	h'i:	w'i:	2	0.500

In the IBM Watson Text-to-Speech to Sphinx experiment, the number of illegal-legal word conversions identified was 2, and the overall word error rate was 65.5%. The illegal-legal word conversions and the corresponding Levenshtein distances and ratios were as follows:

original	transcribed-illegal	espeak_phonemes-original	espeak_phonemes-transcribed	Levenshtein-orthographic	orthographic-ratio
hours	ours	'aU@z	'aU@z	1	0.889
years	yours	j'i@3z	j'o@z	4	0.600

In the IBM Watson Text-to-Speech to IBM Watson Speech-to-Text experiment with the small Attempto lexicon, the number of illegal-legal word conversions identified was 5, and the overall word error rate was 24.6%. The illegal-legal word conversions and the corresponding Levenshtein distances and ratios were as follows:

original	transcribed-illegal	espeak_phonemes-original	espeak_phonemes-transcribed	Levenshtein-orthographic	orthographic-ratio
buy	I	b'aI	_#X1'aI	4	0
by	I	b'aI	_#X1'aI	3	0
eye	I	'aI	_#X1'aI	4	0
eyes	I	'aIz	_#X1'aI	5	0
hour	our	'aU@	'aU3	1	0.857

In the IBM Watson Text-to-Speech to IBM Watson Speech-to-Text experiment with the large Attempto lexicon, the number of illegal-legal word conversions identified was 34 (excluding duplicate results from experiment 4), and the overall word error rate was 44.8%. The illegal-legal word conversions and the corresponding Levenshtein distances and ratios were as follows:

original	transcribed-illegal	espeak_phonemes-original	espeak_phonemes-transcribed	Levenshtein-orthographic	orthographic-ratio
puny	any	pj'u:ni	'Eni	3	0.571
paeony	any	p'i:;@nI	'Eni	3	0.667
peony	any	p'i@ni	'Eni	4	0.500
tinny	any	t'Ini	'Eni	4	0.500
pinny	any	p'Ini	'Eni	4	0.500
enmity	anybody	'EnmI2t#i	'EnI2b,0di	9	0.308
bide	I	b'aId	_#X1'aI	5	0
thigh	I	T'aI	_#X1'aI	6	0
bye	I	b'aI	_#X1'aI	4	0
bine	I	b'aIn	_#X1'aI	5	0
aye	I	'aI	_#X1'aI	4	0
chive	I	tS'aIv	_#X1'aI	6	0
mi	me	m'aI	m'i:	2	0.500
knee	me	n'i:	m'i:	4	0.333
nee	me	n'i:	m'i:	3	0.400
née	me	n'i:	m'i:	3	0.400
mime	mine	m'aIm	m'aIn	2	0.750
cower	our	k'aU3	'aU3	4	0.500
arse	ours	'A@s	'aU@z	4	0.500
threes	these	Tr'i:z	D'i:z	3	0.727
leys	these	l'eIz	D'i:z	5	0.444
bees	these	b'i:z	D'i:z	5	0.444
beefs	these	b'i:fs	D'i:z	6	0.400
lees	these	l'i:z	D'i:z	5	0.444
zees	these	z'i:z	D'i:z	5	0.444
biz	this	b'Iz	D'Is	5	0.286
wee	we	w'i:	w'i:	1	0.800
weedy	we	w'i:di	w'i:	3	0.571
weep	we	w'i:p	w'i:	2	0.667
weal	we	w'i:l	w'i:	2	0.667
gooey	we	g'u:I	w'i:	5	0.286
reap	we	r'i:p	w'i:	4	0.333
wreath	we	r'i:T	w'i:	4	0.500
wheel	we	w'i:l	w'i:	3	0.571

For this last experiment, an expected value for the number of legal-illegal word misrecognitions occurring by chance based on the binomial distribution was calculated of 3.9. This was the result of multiplying the number of 'trials' (equivalent to the number of potential 'adversarial' words tested i.e. 77,689) by the probability of 'success' ('success' from the attacker's point of view being a legal-illegal word misrecognition, as opposed to a word match or a legal-legal word misrecognition). The success probability was calculated using an estimated number for the word recognition capacity of the speech recognition systems used in the experiments of 134,000. This

number was based on the number of entries in the pronunciation dictionary CMUdict as of November 2014[8]. The success probability was thus calculated as the multiple of the overall error rate for the experiment (44.8%) and the probability of recognising an Attempto illegal word by chance in the estimated word space (15/134,000), resulting in a success probability of 0.00005%. Multiplying this probability by the number of trials (77,689) resulted in the expected value of 3.9. The probability of the actual observed number of legal-illegal word misrecognitions of 34 was calculated by approximating the binomial distribution to the normal distribution (calculation of the binomial distribution probability was not feasible given the very high number of 'trials'). This involved applying a continuity correction factor for continuous distributions to the observed value, resulting in an observed value of 33.5, then subtracting from the observed value the expected value of 3.9, and then dividing the difference by the standard deviation (calculated as the square root of the variance, the variance being number of trials (77,689) multiplied by the probability of success (0.00005%) multiplied by the probability of failure (0.99995%), resulting in a variance of 3.88 and a standard deviation of 1.97). The result of this calculation was a z-score of 15.025, indicating a vanishingly small probability of the observed result having been produced by chance.

7 DISCUSSION

Overall the results of the experiments described above confirm the feasibility of a black-box attack on a speech recognition system, based solely on identifying mismatches between input to and output from the system without requiring any detailed knowledge of the system's inner workings. These results represent a motivation for future work on possible methods for improving the defences of speech recognition systems against this kind of attack. With regard to individual findings, one noteworthy finding is the presence in the legal-illegal word conversion results of pairs of words which are very dissimilar in terms of their orthographic edit distance, with orthographic Levenshtein ratios of below 0.5 (chosen as an arbitrary cut-off point). The fact that one of the words in these in these pairs is nevertheless recognised as the other word supports the possibility of covert attacks on a speech-controlled system using words which would not appear to the victim to be related to the attack. Examples of pairs of very dissimilar words in the results in terms of their orthographic edit distance are 'boss/us', 'exactly/we', 'bide/I' and 'enmity/anybody'. As regards the phonemic strings, the most noteworthy finding is the presence in the illegal-legal word conversion results of original-transcribed word pairs with identical phoneme strings, indicating the presence of a homophone pair, i.e. a pair of words with different spellings and meanings but identical pronunciation. Instances of this in the experiment results include 'hours/ours' and 'wee/we'.

The high overall word error rate in all of the experiments is likely to be related to the nature of the input as individual words rather than sentences. The language models incorporated in speech recognition systems assess the likelihood of words occurring in a specific context, which is not possible for individual words in isolation. Thus the word error rate might be expected to be higher for a list of unconnected words than for sentences or longer texts.

Another result likely to be related to a lack of linguistic context is the significant degradation of the performance of the Watson-to-Watson combination from a 24.6% word error rate in the fourth experiment with the small Attempto lexicon to a 44.8% word error rate in the fifth

[8] See http://www.speech.cs.cmu.edu/cgi-bin/cmudict

experiment with the large Attempto lexicon data. It can be speculated that this difference in performance is related to a likely higher proportion of less common words in the Attempto large lexicon than in the small one; i.e. in the absence of any context for individual words, the Watson language model will be biased towards identifying a speech sound as a word which is more common in the language in general. Thus many of the relatively uncommon words in the Attempto large lexicon may be being misrecognised a more common word, on account of the language model bias.

The finding of legal-illegal words conversions occurring at a rate far higher than random chance in the last experiment is also likely to be due to a language model bias. The words declared illegal in Attempto are very common pronouns and determiners in natural English. As stated above, in the absence of linguistic context a language model is likely to have a bias towards recognising speech sounds as words which are frequent in the language in general, thus common words will have a higher probability of being misrecognised than less common ones.

These results lead to a more general suggestion that the processing of linguistic context by the language model may have contradictory effects on speech recognition performance. Whilst the language model processing may improve the recognition of words in familiar contexts, it may have the opposite effect where a word is presented in an unfamiliar context; i.e. based on the language model, the system may misrecognise a speech sound as a word which it expects to be present in the given context rather than as the word that was actually spoken, even if the acoustic model identifies a conflicting phone sequence. Based the experimental results of this project, it appears that far from strengthening the word recognition capacity of a speech recognition system, the language model may in fact be the root cause of word misrecognition in many instances.

The above notwithstanding, one result which cannot be explained by language model bias is the significant difference between the word error rate for the Watson Text-to-Speech to Watson Speech-to-Text experiment with the Attempto small lexicon and the three other experiments with different system combinations (24.6% as compared to 65.5%, 53.0% and 63.7% for the other three system combinations). As the four experiments all use the same data, the superior performance of the Watson-to-Watson system combination as compared to the other system combinations cannot be explained by a bias of the language model towards common words in the absence of linguistic context. The superior performance of the Watson-to-Watson combination cannot either be explained as due to superior quality of the Watson speech synthesizer or of the Watson speech recognizer in themselves, as the experiments in which the two Watson systems are linked to Sphinx and eSpeak respectively do not show a significantly different word error rate to the eSpeak-to-Sphinx experiment. The improvement in word error rate the Watson-to-Watson experiment must rather be due in some way to the quality of the interaction between the input and output systems.

8 FUTURE WORK

There are various possibilities for future work which might follow from the findings of this project. Most obviously, the methodology developed in this project could be applied to other datasets. The experiments performed in this project were based on artificial data, the 'target' words being a list of words declared illegal in Attempto Controlled English. Future experimental work might instead test lists of words used to control a real system. The 'target' words might for

example be command word lists for digital assistants such as Siri and Cortana, vocabularies of artificial languages used for robot control such as the ROILA language presented by Mubin et al., or any set of command words used to control military, medical, or transport technology. Such lists of real-life 'target' words might be tested against a large corpus representative of an entire language, such as the British National Corpus. Alternatively, a set of potential 'adversarial' words might be developed synthetically, by mimicking the natural processes of language shift over time and place. Hay et al. (2015) describe for example a shift in vowel sounds in the evolution of New Zealand English, leading to a situation in which "NZE bat often sounds to speakers from elsewhere like bet, NZE bet sounds like bit, and NZE bit often sounds like but." It might be possible to gradually change a synthesized word sound to identify the point at which the word is misrecognised by a speech recognition system as a different word. A further possibility with regard to the set of 'adversarial' words might be to consider, beyond individual words, possible phonetic overlaps between neighbouring words in continuous speech. Kaljurand and Alumäe (2012) note the potential confusion arising from a lack of clear boundaries between words in spoken language (eg. 'I scream' and 'ice cream' are difficult to distinguish in hearing). Lastly, it might also be possible to consider phonetic overlaps between speech sounds in different languages. Phonetic overlap between different languages raises the possibility of an attack on a speech recognition system using not only apparently unrelated words, but words spoken in one language which may be recognised by a speech recognition system as words spoken in another.

Future work might further consider measures for minimising misrecognition of 'target' words as exemplified by the experiments performed in this project. Such work might focus on improving the ability of speech recognition systems to detect and avoid word misrecognition. Huang et al. (2014) discuss the current inability of speech recognition systems to detect conflicts between the acoustic model and language model speech processing results which might indicate the presence of a word recognition error, and claim a need "to create systems that reliably detect when they do not know a (correct) word." The results of this project confirm a need for further work in this regard. A further possibility would be to use the results of this project as a basis for developing a method of selecting command words which have a minimal probability of being subject to misrecognition by a given system. The 'target' words used in the experiments conducted in this project were shown to have a far higher probability than random chance of being substituted for another word. The methodology used in this project might be developed to aid selection of command words which have the opposite property of having a lower than accidental probability of being subject to misrecognition.

9 CONCLUSIONS

This project demonstrated the feasibility in theory of a black-box attack on speech recognition systems. The attack is based on identifying words which are external to the system's vocabulary but which the system may misrecognize as internal vocabulary words. This work was performed in the artificial context of a controlled natural language, seeking to identify instances where a word which is part of the controlled language vocabulary is misrecognized as a word which is not permitted in the controlled language. Several instances of conversion of a 'legal' word to an 'illegal' word in two different speech recognition systems were found. In a real-life context, an attacker might seek to find instances of conversion from a word apparently unrelated to control of a system to a word which the system recognizes as a command, such as a word in a list of commands for a digital assistant. A broad result from this work is that security testing of speech

recognition systems needs to consider not only the recognition accuracy of words which are intended to be recognised by the system, but also the space of speech sounds which are not intended to be recognised by the system, but which may be misrecognised as one of the words of which recognition is intended. The space of speech sounds to be considered includes potentially not only individual words, but also overlap between words in continuous speech, as well as phonetic overlaps between speech sounds in different languages. Thus there is much scope for future work.

REFERENCES

Barber, Daniel, Ryan W. Wohleber, Avonie Parchment, Florian Jentsch, and Linda Elliott. "Development of a Squad Level Vocabulary for Human-Robot Interaction." In *International Conference on Virtual, Augmented and Mixed Reality*, pp. 139-148. Springer International Publishing, 2014.

Carlini, Nicholas, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. "Hidden Voice Commands." In *25th USENIX Security Symposium (USENIX Security 16), Austin, TX*. 2016.

Craparo, Emily, and Eric Feron. "Natural language processing in the control of unmanned aerial vehicles." In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, United States*. 2004.

Cutajar, Michelle, Edward Gatt, Ivan Grech, Owen Casha, and Joseph Micallef. "Comparative study of automatic speech recognition techniques." *IET Signal Processing* 7, no. 1 (2013): 25-46.

Dhanjani, Nitesh. *Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts*. O'Reilly Media, Inc., 2015. [Chapter 8: Hearing Voices]

Fuchs, Norbert E., and Rolf Schwitter. "Attempto controlled english (ace)." *arXiv preprint cmp-lg/9603003* (1996).

Hay, Jennifer B., Janet B. Pierrehumbert, Abby J. Walker, and Patrick LaShell. "Tracking word frequency effects through 130 years of sound change." *Cognition* 139 (2015): 83-91.

Hinton, Geoffrey, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." *IEEE Signal Processing Magazine* 29, no. 6 (2012): 82-97.

Huang, Xuedong, James Baker, and Raj Reddy. "A historical perspective of speech recognition." *Communications of the ACM* 57, no. 1 (2014): 94-103.

Kaljurand, Kaarel, and Tanel Alumäe. "Controlled natural language in speech recognition based user interfaces." In *International Workshop on Controlled Natural Language*, pp. 79-94. Springer Berlin Heidelberg, 2012.

Lison, Pierre, and Raveesh Meena. "Spoken dialogue systems: the new frontier in human-computer interaction." *XRDS: Crossroads, The ACM Magazine for Students* 21, no. 1 (2014): 46-51.

Mehrabani, Mahnoosh, Srinivas Bangalore, and Benjamin Stern. "Personalized speech recognition for Internet of Things." In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pp. 369-374. IEEE, 2015.

Mubin, Omar, Christoph Bartneck, and Loe Feijs. "Towards the design and evaluation of roila: a speech recognition friendly artificial language." In *International Conference on Natural Language Processing*, pp. 250-256. Springer Berlin Heidelberg, 2010.

Mubin, Omar, Joshua Henderson, and Christoph Bartneck. "You just do not understand me! Speech Recognition in Human Robot Interaction." In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 637-642. IEEE, 2014.

Nathan, Cherie-Ann O., Vinaya Chakradeo, Kavita Malhotra, Horacio D'Agostino, and Ravish Patwardhan. "The voice-controlled robotic assist scope holder AESOP for the endoscopic approach to the sella." *Skull Base* 16, no. 03 (2006): 123-131.

Oehlert, Peter. "Violating assumptions with fuzzing." *IEEE Security & Privacy* 3, no. 2 (2005): 58-62.

Padmanabhan, Mukund, and Michael Picheny. "Large-vocabulary speech recognition algorithms." *Computer* 35, no. 4 (2002): 42-50.

Papernot, Nicolas, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. "Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples." *arXiv preprint arXiv:1602.02697* (2016).

Pucher, Michael, Andreas Türk, Jitendra Ajmera, and Natalie Fecher. "Phonetic distance measures for speech recognition vocabulary and grammar optimization." In *3rd Congress of the Alps Adria Acoustics Association*, pp. 2-5. 2007.

Qin, Long. "Learning out-of-vocabulary words in automatic speech recognition." PhD diss., Carnegie Mellon University, 2013.

Schalkwyk, Johan, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope. "“Your Word is my Command”: Google Search by Voice: A Case Study." In *Advances in Speech Recognition*, pp. 61-90. Springer US, 2010.

Appendix A: SOURCE CODE

Files:

- Clex_processing.py *(used for randomized chunking of the Attempto large lexicon)*
- schayn_espeak2sphinx.py *(used for small Attempto lexicon experiment)*
- schayn_espeak2watson.py *(used for small Attempto lexicon experiment)*
- schayn_watson2sphinx.py *(used for small Attempto lexicon experiment)*
- schayn_watson2watson.py *(used for small Attempto lexicon experiment)*
- schayn_watson2watson0.py *(used for large Attempto lexicon experiment)*
- espeak_phonemes.py *(used to generate eSpeak phoneme strings)*
- Levenshtein.py *(used to calculate Levenshtein distances and ratios between original and transcribed words)*

NB The programs used for the small Attempto lexicon experiments included code for calculating the Levenshtein distances and ratios between literal eSpeak phoneme strings for original and transcribed words. However, the results of these calculations were found not to be meaningful in terms of measuring phonetic distance, and were therefore not included in the results (the distance was overestimated in most instances, due to the presence of special characters in phonemic strings, as well as to the representation of single phonemes by symbols of differing lengths, as detailed in Appendix D).

Appendix B: THIRD PARTY SOFTWARE

Python v.3.5.2

Python Package Index module SpeechRecognition v.3.4.6

pocketsphinx-python v.0.1.0 (wheel package for PyPI SpeechRecognition module)

swigwin v.3.0.10 (Swig for Windows, dependency for pocketsphinx-python)

eSpeak v.1.48.04

python-espeak (https://github.com/relsi/python-espeak)

Natural Language Toolkit v.3.2.1 (Python Package Index module)

Watson-developer-cloud v.0.18.1 (Python Package Index module)

IBM Watson Text-to-Speech https://www.ibm.com/watson/developercloud/text-to-speech.html

IBM Watson Speech-to-Text https://www.ibm.com/watson/developercloud/speech-to-text.html

Python code for calculating Levenshtein distance from rosettacode.org (incorporated in schayn.py)

Appendix C: DATA

Attempto illegal words (15 words extracted from illegalwords.pl at https://github.com/Attempto/APE/tree/master/lexicon)

Attempto content and function words (1,214 words extracted from clex_lexicon.pl and functionwords.pl at https://github.com/Attempto/APE/tree/master/lexicon. NB some overlap between the function word list and the content word list included in this list was identified):

a, abaci, abacus, abacuses, aboard, about, above, absinth, accept, accepted, accepts, access, accesses, account, accounts, across, active, address, addressed, addresses, admissible, aerie, aeries, aery, after, against, age, aged, ages, aircraft, airline, airlines, all, allow, allowed, allows, along, alongside, alto, altos, always, amid, among, amongst, an, ancestor, ancestors, and, angrier, angriest, angry, animal, animals, anon, ape, aped, apes, appear, appears, apple, apples, approach, approached, approaches, approximate, approximated, approximates, are, aren, around, arrive, arrives, article, articled, articles, as, as-soon-as-possible, ask, asks, asset, assets, assign, assigned, assigns, associated, assume, assumed, assumes, at, authenticate, authenticated, authenticates, automatic, automatics, average, averaged, averages, await, awaited, awaits, bad, ball, balls, bank, banked, banks, bark, barked, barks, be, beat, beaten, beats, bed, bedded, beds, beer, beers, before, behind, belie, belied, belies, believe, believed, believes, belong, belongs, below, ben, beneath, bens, beside, best, better, between, beyond, big, bigger, biggest, bike, biked, bikes, bite, bites, bitten, blame, blamed, blames, blink, blinks, blue, blued, bluer, blues, bluest, bodies, body, boil, boiled, boils, bone, bones, book, booked, books, boss, bosses, bought, box, boxed, boxes, boy, boys, branch, branches, bretheren, bring, brings, brother, brothers, brought, but, button, buttoned, buttons, buy, buys, by, cake, cakes, can, cancel, cancelled, cancels, canned, cans, car, card, carded, cards, carefully, carnivore, carnivores, carried, carries, carry, cars, case, cased, cases, cashier, cashiered, cashiers, cat, cats, cave, caves, cellular, charge, charged, charges, chase, chased, chases, check, checked, checks, cheese, cheeses, child, children, circle, circled, circles, circumference, circumferences, cities, city, clean, cleaned, cleaner, cleanest, cleans, clerk, clerks, clever, cleverer, cleverest, code, coded, codes, cold, colder, coldest, colds, collapse, collapsed, collapses, color, colors, come, come-from, comes, comes-from, companies, company, computer, computers, consider, considered, considers, contain, contained, contains, content, contented, contents, contract, contracted, contracts, correct, corrected, corrects, count, counted, countries, country, counts, cover, covered, covers, cow, cowed, cows, credential, criminal, criminals, customer, customers, dance, danced, dances, database, databases, date, dated, dates, day, days, declaration, declarations, deep, deeper, deepest, deeps, deliver, delivered, deliveries, delivers, delivery, description, descriptions, desk, desks, despite, development, developments, diameter, diameters, dirtied, dirtier, dirties, dirtiest, dirty, display, displayed, displays, do, doctor, doctored, doctors, doe, does, doesn, dog, dogged, dogs, don, donkey, donkeys, down, download, downloaded, downloads, dozen, drink, drinkable, drinks, drive, driven, drives, drop, dropped, drops, drunk, dummies, dummy, during, each, ease, eased, eases, easily, eat, eaten, eats, egg, eggs, eight, elder, eldest, element, elements, eleven, emptied, empties, empty, enormous, enter, entered, enters, error, errors, evening, evenings, every, everybody, everyone, everything, evil, evils, exactly, exist, exists, expensive, expire, expired, expires, extremely, eye, eyed, eyes, eyrie, eyries, eyry, fair, fairer, fairest, fairs, false, falser, falsest, famous, far, farmer, farmers, farther, farthest, fast, fasted, faster, fastest, fasts, father, fathered, fathers, fear, feared, fears, fed, feed, feeds, female, females, fill-in, fills-in, fitfully, five, flat, flats, flier, flies, fliest, flow, flower, flowered, flowers, flown, flows, fly, fond-of, fonder-of, fondest-of, food, foods, for, form, formed, forms, four, fox, foxed, foxes, fridge, fridges, friend, friends, from, furniture, further, furthest, gale, gales, garden, gardens, get, gets, girl, girls, give, given, gives, go, goal, goals, goat, goats, goes, gone, good, goods, got, gotten, grass, grassed, grasses, grate, grated, grates, great, greater, greatest, green, greener, greenest, greens, group, grouped, groups, grown-up, grown-ups, had, hand, handed, hands, happen, happens, happier, happiest, happily, happy, hard, harder, hardest, has, hate, hated, hates, have, he, hear, heard, hears, height, heights, held, her, hero, heroes, herself, him, himself, his, hit, hits, hold, holds, horse, horses, hour, hours, house, housed, houses, how, human, humans, hungry, hurried, hurries, hurry, husband, husbanded, husbands, id, if, implied, implies, imply, important, impossible, in, insert, inserted, inserts, inside, institution, institutions, instruction, instructions, integer, integers, interest, interested, interested-in, interests, into, invalid, invalids, invite, invited, invites, is, isn, it, its, itself, keep, keeps, kept, know, known, knows, large, larger, largest, laundries, laundry, least, less, life, lift, lifted, lifts, like, liked, likes, list, listed, lists, live, live-at, lived, lives, lives-at, living, livings, long, longer, longest, look, look-at, looks, looks-at, lose, loses, lost, loudly, love, loved, loves, lunch, lunches, machine, machined, machines, mad-about, madder-about, maddest-about, mail, mailed, mails, man, manager, managers, manned, mans, manually, many, master, mastered, masters, mat, mate, mated, mates, mats, matted, may, meal, meals, meat, meats, meet, meets, member, members, men, merchant, merchants, message, messages, met, mice, milk, milked, milks, money, monies, more, morning, mornings, most, mother, mothered, mothers, mouse, mouses, move, moved, moves, much, must, musts, name, named, names, natural, naturals, near, necessary, need, needed, needs, new, newer, newest, nice, nicer, nicest, nine, no, nobody, noone, not, nothing, null, numb, numbed, number, numbered, numbers, numbest, numbs, numeric, object, objects, of, off, offer, offered, offers, office, offices, often, old, older, oldest, on, one, only, onto, open, opened, opens, or, out, outside, over, own, owned, owner, owners, owns, paid, paper, papered, papers, parent, parents, park, parked, parks, password, passwords, past, patiently, patricide, patricides, pay, pays, pencil, pencils, percent, permit, permits, permitted, person, persona, persona-non-grata, personae-non-grata, personal, personal-code, personal-codes, personals, personas, persons, pet, pets, petted, pizza, pizzas, place, placed, places, play, played, plays, point, pointed, points, possible, price, priced, prices, process, processed, processes, program, programmed, programs, proposition, propositioned, propositions, provable, provably, prove, proved, proven, proves, public, publics, queen, queened, queens, quick, quicker, quickest, quickly, rat, ratio, ratios, rats, ratted, raw, read, reads, real, reals, reason, reasons, recommended, red, redder, reddest, reds, referee, refereed, referees, reject, rejected, rejects, relate-to, relates-to, rental, rentals, replace, replaced, replaces, resource, resources, rice, rich, richer, richest, rook, rooked, rooks, room, roomed, rooms, rot, rots, rotted, run, runs, s, sad, sadder, saddest, safely, said, sand, sanded, sands, sat, say, says, school, schooled, schools, score, scored, scores, screen, screened, screens, see, seen, sees, sell, sells, send, sends, sent, sentence, sentenced, sentences, serve, served, serves, service, serviced, services, set, sets, seven, she, sheep, should, sign, signed, signs, silently, sink, sinks, sister, sisters, sit, sits, six, size, sized, sizes, sleep, sleeps, slept, slot, slots, slotted, small, smaller, smallest, smalls, smart, smarted, smarter, smartest, smarts, smell, smelled, smells, smelt, smile, smiles, snore, snores, soften, softened, softens, sold, some, somebody, someone, something, soundly, space, spaced, spaces, sped, speed, speeded, speedily, speeds, station, stations, steal, steals, stolen, street, streets, string, strings, subscription, subscriptions, succeed, succeeded, succeeds, such, sugar, sugared, sugars, sunk, sunken, surface, surfaced, surfaces, syntactic, table, tabled, tables, tail, tailed, tails, take, taken, takes, talk, talked, talks, tall, taller, tallest, tell, teller, tellers, tells, temperature, temperatures, ten, term, terms, test, tested, tests, text, texts, than, that, the, their, them, themselves, then, there, they, thing, things, three, through, throughout, till, tire, tired, tires, title, titles, to, told, ton, tons, toward, towards, town, towns, train, trained, trains, transmit, transmits, transmitted, tried, tries, true, truer, trues, truest, try, twelve, two, type, typed, types, uncle, uncles, under, understand, understands, understood, unnecessary, until, up, upon, use, used, user, users, uses, usually, vaguely, valid, value, valued, values, varied, varies, vary, vehicle, vehicles, via, village, villages, visa, visa-card, visa-cards, visas, visit, visited, visits, wait, wait-for, waited, waits, waits-for, walk, walked, walks, want, wanted, wants, warm, warmed, warmer, warmest, warms, wash, washed, washes, watch, watched, watches, water, watered, waters, wealth, wet, wets, wetted, wetter, wettest, what, when, where, which, white, whiter, whites, whitest, who, whose, wife, win, wine, wines, wins, wisely, with, within, without, wives, wolf, wolfed, wolfs, wolves, woman, women, won, work, work-at, worked, works, works-at, worse, worst, write, writes, written, wrought, year, years, you, young, younger, youngest, your, yourself, yourselves, zero, zip, zip-code, zip-codes, zipped, zips

Attempto content and function words (77,689 words in 1,000 word random chunks from clex_lexicon.pl at https://github.com/Attempto/Clex and functionwords.pl at https://github.com/Attempto/APE/tree/master/lexicon)

Appendix D: eSpeak phoneme symbols

eSpeak uses a system for representing the International Phonetic Alphabet (IPA) in ASCII-encodable characters which was developed by Evan Kirshenbaum (2001). The eSpeak symbols for English phonemes are listed below.

The eSpeak symbol set also includes a number of special characters used to represent non-phonetic aspects of speech. Syllable stress is marked with ' . Word boundaries are marked with _ . Some alphabetic characters may also be used as special characters; eg. “X” can be used to mean “There is no vowel until the word boundary.” Alphabetic special characters are preceded by #.

English Vowels

[@] alph**a** schwa
[3] bett**er** rhotic schwa. In British English this is the same as [@], but it includes 'r' colouring in American and other rhotic accents. In these cases a separate [r] should not be included unless it is followed immediately by another vowel.
[3:] n**ur**se
[@L] simp**le**
[@2] the Used only for "the".
[@5] to Used only for "to".
[a] tr**a**p
[aa] b**a**th This is [a] in some accents, [A:] in others.
[a#] **a**bout This may be [@] or may be a more open schwa.
[A:] p**al**m
[A@] st**ar**t
[E] dr**e**ss
[e@] squ**are**
[I] k**i**t
[I2] **i**ntend As [I], but also indicates an unstressed syllable.
[i] happ**y** An unstressed "i" sound at the end of a word.
[i:] fl**ee**ce
[i@] n**ear**
[0] l**o**t
[V] str**u**t
[u:] g**oo**se
[U] f**oo**t
[U@] c**ure**
[O:] th**ou**ght
[O@] n**or**th
[o@] f**or**ce
[aI] pr**i**ce
[eI] f**a**ce
[OI] ch**oi**ce
[aU] m**ou**th
[oU] g**oa**t
[aI@] sc**ie**nce
[aU@] h**our**

English Consonants

[p] [b]
[t] [d]
[tS] **ch**urch [dZ] **j**udge
[k] [g]
[f] [v]
[T] **th**in [D] **th**is
[s] [z]
[S] **sh**op [Z] plea**s**ure
[h]
[m] [n]
[N] si**ng**
[l] [r] **r**ed (Omitted if not immediately followed by a vowel).
[j] **y**es [w]

Appendix E: GitLab Repository

Full details of all data, code and results for this project are stored at https://gitlab.com/miniproject2/wordmisrecognition (access restricted).