CCAD: the Call Contents Automatic Differentiator

Introduction

In this paper, we describe CCAD, the Call Contents Automatic Differentiator, a naive system for extracting post-cut-through dialing digits while excluding post-cut-through content digits. The system runs with an expected accuracy of 99.4 percent and 98.3 percent in the worst case. Such a system is critical in differentiating so-called “envelope information” (which may legally be collected without a warrant) from “content information” (which may not). This paper describes the algorithm used. Though the specific algorithm is unique, it combines well understood algorithms in an intuitive manner.

We will first discuss the history and technology of the telephone. Then we will detail assumptions made about the audio input to the algorithm and detail the algorithm itself, followed by a description of test methodology. Next, we discuss the results and possible improvements to the algorithm, were it to be deployed in a real-world environment. Finally, we conclude with a summary of findings.

Background

In modern society, we often dial telephones but rarely think about what is required to connect a telephone call. This section explores this topic, as well as some telephonic and digital signals processing (DSP) history.

To start at the highest level, the network used to connect one telecommunications user to another is the PSTN (Public Switched Telephone Network). The first deployment of what would eventually become the PTSN was Bell Telephone Company's, in 1878 [Gast 2001]. Dialing methods have evolved and adapted as the network has grown and as technology has advanced. Initially, operators were required to connect every call - manually plugging short lengths of cable to connect different “circuits” (essentially creating a point to point telephone line). Later, rotary dialing was introduced as a way to automate dialing and reduce the number of operators required. In 1960, the first paper on DTMF (Dual-Tone Multi-Frequency) dialing was published in the Bell System Technical Journal [Schenker 1960]. The first introduction of DTMF to the PTSN happened on November 18, 1963 [Fox 2013]. With minor variation, DTMF has remained the standard since. Today, the DTMF standards are detailed in the International Telephone Union's recommendations Q.22, Q.23 and Q.24. Recently, much of the phone system has been digitized, but the user-facing interface (DTMF) has remained the same.

DTMF at its core is a set of eight tones (four high and four low) [ITU-T Q.23 1988]. Each pair of tones (one from the high set, one from the low set) conveys one of the signals 0 through 9, A through D, the star and the octothorp (“pound sign”). Though the signals A through D never made it into mainstream use, they remain in the standard.

DTMF decoding software has been around since the origin of Digital Signals Processing (DSP). The oldest freely-available paper on implementing DTMF detection in software we can locate is from 1989 [Mock 1989]. However, we are confident this is not the first software implementation – if nothing else, there would have been proprietary implementations. Paper [Chen 1996] after paper [Clarkson 2004]

describing various implementations has followed, as have open-source implementations [Blue 1997] [Zapata 2001][Digium 2002].

The algorithm also performs Voice Activation Detection (VAD) – determining which parts of audio contain a person speaking and which contain noise. VAD is an area of ongoing research. However, this paper relies only on one of the many measures used in VAD [Sahidullah 2012] – energy detection. Energy detection simply calculates the average volume of section of speech and is the most obvious and most simple possible measure for whether or not there is speech in an audio stream. However, it performs extremely poorly if the environment is noisy.

Assumptions

This system makes several reasonable assumptions about the format of a call. First, it assumes that the call it is examining is a user calling an automated system. The canonical example for this software is an international calling service – the user calls in, the service requests subscriber information and the number to be called. Other examples of automated systems include the service lines of banks and other financial institutions. This assumption implies a "call-and-response" style interaction. That is, it assumes that after the call is connected to the initial recipient (callee), information is requested from the caller via a pre-recorded voice prompt (e.g. "Press 1 for English, Press 2 for Spanish...", "Please enter your account number, followed by the pound sign"). The user then responds to this prompt by pressing a button or buttons on their telephone. This process then repeats until the automated service has collected the information it requires.

This system assumes a very minimal PSTN system. It assumes that DTMF is transmitted via the same channel as the voice and would be audible to any person listening to the call. It also assumes that a single audio stream contains audio from both the caller and the callee. This renders the requirements so simple that this system would work with any relatively modern telephone system, and the technical requirements are met in parts of the US phone system back to the first DTMF deployments and are certainly met by all of the phone system today.

Finally, this system expects that the input audio stream does not begin until after the call has been connected. That is, this assumes it does not receive the originally dialed ten-digit phone number.

The Algorithm

This system uses two-stage process to determine which portions of the audio stream constitute envelope information. Stage 1 is the extraction of a "signal stream" from the audio, containing all DTMF signals and all separators. Stage 2 examines the signal stream using simple heuristic filters to determine what actually is envelope information. The final output of this methodology is any envelope information that was embedded in the audio stream.

Stage 1 of the method extracts DTMF information and timing information from the audio stream. Timing information is the length of any silences or voice in the audio, and is used to separate DTMF digits into meaningful groups. By default, this implementation considers voice longer than one second

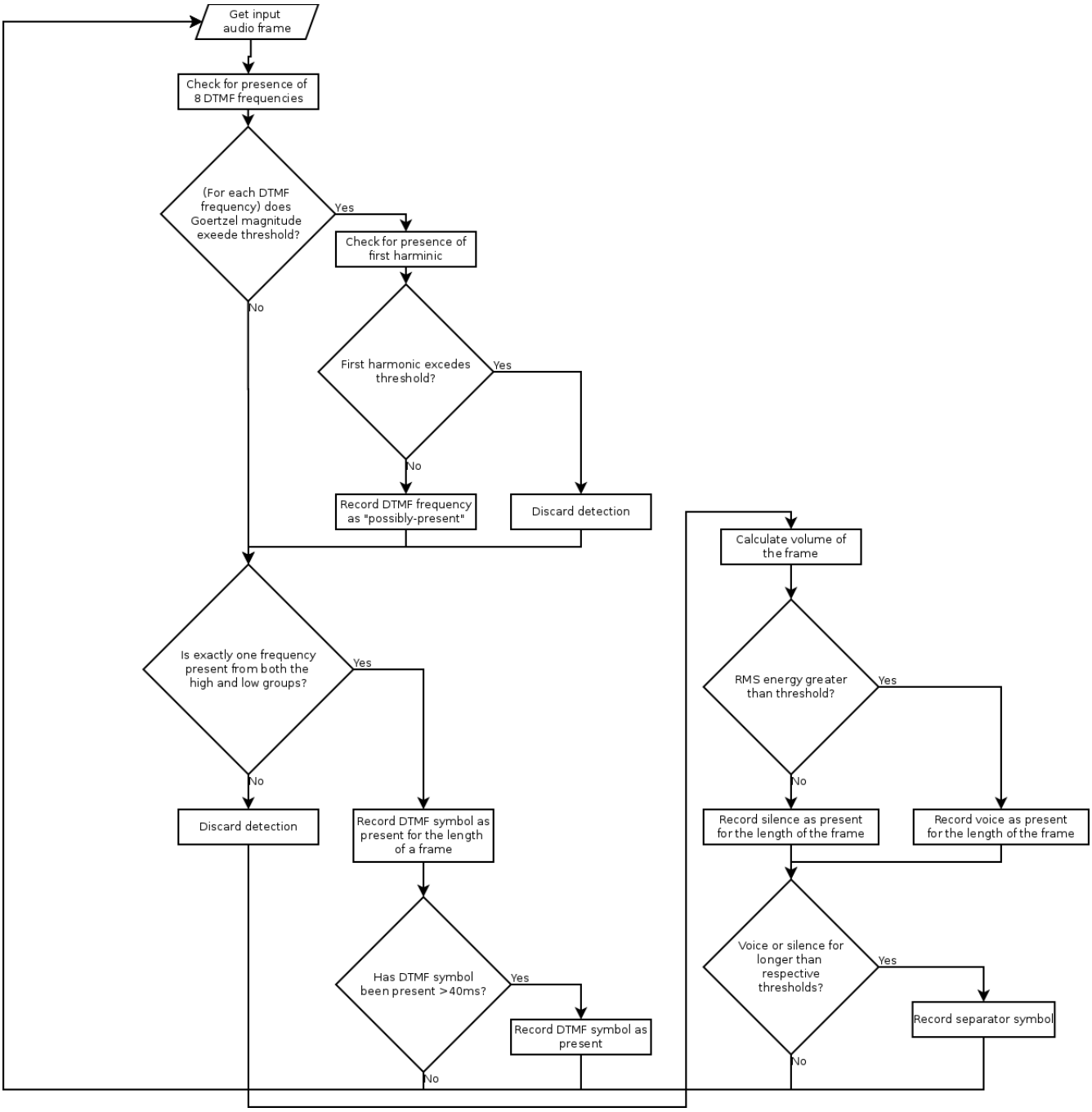
or silence longer than ten seconds to constitute a separator between digit groups. In order to do this, we must first differentiate between audio that contains DTMF signals and that which does not. Then, for audio which does not contain DTMF tones, we must discern between voice audio and silence audio.

Extraction of the DTMF signals is a well understood problem. Technical documentation is available going back to the 1980s [Mock 1989] describing (or containing) programs for doing so. The most popular (and simplest to implement) method is the Goertzel Algorithm [Goertzel 1958], which can be used to determine if a specific frequency band is present. Hobbyists have implemented DTMF signal recognition as early as 1997 [Blue 1997] and it is used in leading open-source software [Digium 2002].

The Goertzel Algorithm is applied to the eight DTMF frequencies individually. For each frequency, the output of the Goertzel Algorithm (a unitless magnitude) is compared to a pre-set threshold. If the output is greater than the threshold, the corresponding DTMF frequency might be present. Next, for every frequency which was greater than the threshold, the first harmonic (frequency twice the original) is checked. DTMF tones are mechanically generated and will not have any output at the first harmonic. In contrast, voice or non-mechanical sounds will have a first harmonic. Therefore, if a first harmonic is detected, the DTMF frequency detection is a false positive and is ignored.

Next, the set of DTMF frequencies detected is examined to make sure that exactly two are present – one from the high set, one from the low. If more than one tone from a set is present, or a set has no tones present, the detection is a false positive and is ignored. If the frequency set passes this test, it is a potential DTMF signal.

Finally, the length of time the DTMF signal has been present is measured. ITU-T Q.23 requires DTMF signals be present for a minimum of 40 milliseconds to be valid, so any shorter signals are ignored. Any potential signal which passes this test is a valid signal and is added to the signal stream.



Drawing 1: Logical flow of CCAD Stage 1

If a section of audio does not contain DTMF tones, we must then determine if it is silence or voice content. In order to do this, the most naive possible algorithm is used. We simply measure the volume of the section of audio. If it is above a certain volume, it is voice. If below, silence/noise. The length of each voice and noise section is tracked. If a section of audio contains voice, it is also counted towards the silence length. These lengths are both reset when DTMF signals are detected and the length of voice is reset when silence is detected. If at any point the tracked length of voice goes over one second or the tracked length of silence exceeds ten seconds, a “silence signal” is added to the

signal buffer to act as a separator between sequences of DTMF signals.

In stage two of the algorithm, the DTMF signal buffer is broken into segments. A segment is any series of signals between silence signals. Each segment is then examined for validity as a possible phone number using simple pattern matching. For example, if a sequence of DTMF signals is 10 signals long (or 11 with an octothorp as the final signal) and consists only of zero through nine, it could be a valid US telephone number and is marked as such. On the contrary, a 16-signal DTMF sequence consisting of zero through nine (optionally with the 17th signal as an octothorp), it is not a valid telephone number – more likely a credit card number – and is ignored. As a further example, if a sequence contains A through D, star or an octothorp (with the exception that it may end in an octothorp), it is not a valid telephone number and is ignored.

The implementation accompanying this paper is written to discover domestic (US+Canada) calls only, following the North American Numbering Plan format [NANPA], though it could easily be expanded to include the full range of international numbers defined by the ITU [ITU-T E.164 2011].

Test methodology

CCAD was tested using a modified set of audio from ITU-T Recommendation P.23's supplemental audio database. The “original” (*.SRC) voice files from this database were preprocessed by removing any sections longer than 0.1 second with a volume less than -40dBFS – in short, by removing any silence – followed by adjusting the volume such that the maximum peak amplitude was 0 dBFS – as loud as possible without losing any content. Only the white noise file from the same database was used to provide noise, and was not modified. The modification of voice files was performed to compensate for the naive VAD algorithm.

Tests were performed in two stages. First, a set of semi-random audio streams was generated. Second, a set of more restricted format audio streams was generated. Each set consisted of 1 million audio streams. For each generated audio stream, the signal stream (DTMF signals and separators based on voice time and silence time) corresponding to the generated audio was saved. The audio stream was run through the detection algorithm implementation and the results compared to the expected results.

The first set of audio streams consist of randomized sequences of DTMF signals, voice samples and noise samples. No ordering between types of audio was imposed. Voice and noise sections had a minimum length of zero and no maximum while DTMF signals were generated in lengths of 1-16, with no restrictions on signal usage or sequence. This set of input streams was used as a stress test of the DTMF detection and VAD algorithms, determining their accuracy in the worst case. Only the stage 1 output was examined to determine success.

The second set of audio streams was intended to represent more typical inputs. This generated input sequences where a DTMF section was always followed by a voice or silence section exceeding the threshold for separation. This more closely models the call-and-response format assumed by the algorithm under test. Only the stage 2 output was examined to determine success.

In each of these sets of inputs, voice and silence very close (within 100 milliseconds) to their respective time thresholds were shortened or lengthened to be 100 milliseconds to either side of the threshold. This was done to prevent incorrect input files (such as voice input files that contained short silences) from corrupting the test by creating audio sequences which did not match the expected signal sequence.

Performance measurements were gathered on the same machine used to run the million-stream tests. This machine is an Amazon AWS m4.16xlarge machine (64 CPUs, 256 GiB of memory) with an attached 2TiB 20,000 IOPS disk. Additionally, a second, smaller 10,000 stream test was run on a small system to help determine scaling. The smaller system was a Gigabyte C847N-D motherboard with two Intel Celeron 847 processors (running at 1.10GHz) and 2GiB of memory. When this system was built in 2013, it cost less than $125.

Results

Overall, CCAD showed an excellent success rate, especially for such a naive implementation. The type 1 tests (stress tests) showed a success rate of 98.3 percent, while the type 2 (expected conditions) tests showed a 99.4 percent success rate.

In order to better understand other possible improvements, we conducted an examination of the causes of failure for the first 100 failures in each test type. Failures were categorized as one or more of the algorithm having: missed a DTMF signal, missed a separator signal, added an extra DTMF signal, or added an extra separator. Additionally, failures were marked as either benign or not. A benign failure is only applicable to type 1 tests and represents a failure where the generated signal stream was different, but in which the final output (i.e. detected envelope data) would not be different. This category only captures extra or missing separator signals adjacent to other separator signals or adjacent to the start or end of the stream.

	Type 1	Type 1 Benign	Type 2
Missed DTMF	0	-	0
Missed Separator	24 (24.3%)	23 (22.8%)	100 (100%)
Extra DTMF	0	-	0
Extra Separator	40 (39.6%)	14 (13.9%)	0

Table 1: Causes of failure

This failure analysis leads to several interesting results. First, about 37 percent of the examined type 1 failures were benign. Each observed benign failure was at the start or the end of the signal stream. Therefore, these are most likely due to differences in accounting in initial or final conditions between the test generator and the implementation than any actual error. If the ratios of failures held for the larger data set and the benign failures were corrected, the type 1 tests would have an accuracy of 98.9 percent – much more in line with the accuracy of the type 2 tests. Second, all of the type 2 failures were due to missing separators.

Finally, it is worth noting that all the failures are due to VAD issues. This indicates that the DTMF detection and the algorithm for identifying valid phone numbers is extremely robust and reliable. As expected, the VAD algorithm used needs improvement.

It is also worth examining the runtime of these tests to determine what real-world resource usage would be.

	AWS m4.16xlarge	Gigabyte C847N-D
Cores	64	2
Memory (GiB)	256	2
Test size (total type 1&2 streams)	2,000,000	20,000
Test size (total seconds of audio)	245522198.508	2475780.9
Test size (audio, scaled)	466y, 295d, 0:38:30	4y, 258d, 7:00:53
Seconds of audio per core	3836284.3516875	1237890.45
Test runtime (total seconds)	6586.342102	2202.602432
Seconds of audio processed per second per core	582.4605361028	562.0126592142
Minutes of audio processed per second per core	9.7076756017	9.3668776536

Table 2: Performance information

The performance of these two machines – one very high-end and one very low-end - is surprisingly similar – both were able to process about 570 seconds of audio per second per core.

The average call is about two minutes long [Orlowski 2013] and the average person makes 2.5 phone calls per day [Lenhart 2010]. Let's say we wanted to monitor one million people in the US – about 1/3rd of one percent of the population - an absurdly large percentage to suspect of foreign intelligence or terrorist connections. Assuming no calls between citizens on the watch list, this would mean processing 300,000,000 seconds of audio per day.

Let's first look at Amazon-equivalent systems. This requires about 515,000 Amazon processor cores – just over 8,000 Amazon-equivalent systems. Unfortunately, pricing information is unavailable for hardware equivalent to the Amazon systems.

Next, let's examine what it would take to monitor these calls with the Gigabyte system. It would require about 515,000 processor cores, or about 258,000 systems. Fortunately we do know the pricing information for these systems – the hardware for these systems would cost about $32 million. An exceedingly reasonable price, and one that could be significantly reduced by optimizing the hardware for price per core or by improving the performance of the CCAD implementation.

While this is a large number of systems, it is by no means unheard of within supercomputing. Further, many of the typical supercomputing problems (power, cooling) can be sidestepped by distributing these systems throughout the country in local telephone exchanges (coincidentally placing them as close to

the person being monitored as possible). Data transmission and aggregation would be negligible, given that this system reduces large audio streams (kilobytes or megabytes each) to very small strings of text (bytes each).

In short, monitoring a significant portion of telephone calls made within the US is practicable with the implementation of CCAD accompanying this paper – and would become more practicable with an optimized implementation.

Future Work and Possible Improvements

The list of possible improvements to CCAD is significant. This algorithm is an incredibly naive method for performing this test.

Stage one (signal stream detection) can be massively improved by using more advanced algorithms. This implementation used a simplified Goertzel algorithm simply because it was expedient and easy to find reference material on.

In this case, the Goertzel algorithm is not very efficient. The Goertzel Algorithm is useful for checking the presence of a single frequency or a small set. However, the algorithm outlined in this paper tests enough frequencies that it is possible that another algorithm from the same family (DFTs – Discrete Fourier Transforms) may be more efficient.

Additionally, the use of the Goertzel algorithm lead to a “windowing” problem. Because the Goertzel algorithm works on finite, non-overlapping sections of audio, the granularity used for timing is extremely coarse – about 10 milliseconds. For the typical DTMF decoder, which is concerned only with determining if and when there are signals present, this is sufficient. For more advanced versions of this algorithm (which need extremely precise timing information, see later in this section) this granularity is so large as to be unusable. Instead, one could substitute an algorithm from the same family which is either non-windowed or uses a sliding window instead.

Any of the algorithms used for audio processing could easily be adapted to run efficiently on GPUs, allowing thousands of streams to be processed simultaneously and extremely efficiently.

Second, the VAD used in this algorithm is incredibly useless outside controlled settings. Instead, a more complete VAD algorithm could be implemented. VAD is an active field of research within DSP (Digital Signals Processing) as it is extremely useful to any application where a speaker may be silent much of the time. Significant research has been devoted to creating and refining various VAD algorithms.

Alternatively, many modern telephone networks have converted to digital transmission of audio. Network providers are able to significantly reduce the infrastructure required by only transmitting when a person is speaking – so they already perform VAD. This is why, for example, when talking on a phone you may only be able to hear some types of background noise when a person is speaking. With the cooperation of a digital telephony network provider, input audio could come “pre-classified”

as either silence or non-silence.

Stage two can be improved using various methods to infer intent, rather than simply detecting segments that match the pattern of a phone number. The first and most obvious method is to build a database of known phone numbers. This would store both numbers known not to have envelope information sent via DTMF tones (e.g.: banks, credit card companies) and those known to have envelope information sent after the initial call is connected (e.g.: international dialing services). Merely through these two categorizations, the great majority of potential envelope information can be properly categorized.

A second potential improvement is to use the “inter-digit time” - the length of time between tones - to guess intent. For example, US phone numbers are written in groups of 3-3-4 (e.g. 555-867-5309). People tend to dial numbers in the same format they're written, with longer pauses between groups of digits. This may be because they're reading them and it is simpler to remember a small part of the number, or because they dial one portion then and then look for the next part or because they've memorized the number in this format. Other numbers of similar lengths will be broken up differently (e.g credit cards – four groups of four), so the pauses between groups of digits will be placed differently.

A third possible improvement is to use voice recognition software to look for keywords or perform language processing to determine if the user is being asked for envelope information.

All of these methods could be combined. When an audio stream first starts, the algorithm would check if the callee's 10-digit number is a known number stored in the database. Based on this information, it can either ignore the call (if the call will never contain envelope information), or continue listening (if the number is not listed or is known to contain envelope information). If it continues listening, it would then begin transcribing any voice into text and examining it for phrases like “please enter the number you wish to dial”. Concurrently, it would gather precise timing information on the entry of DTMF signals. The presence of keywords and the timing information would be used to weigh whether or not a particular signal sequence likely represents a phone number or not.

Conclusion

In this paper we have described CCAD, the Call Contents Automatic Differentiator, an exceedingly simple and naive system for separating dialed phone numbers, which are routing information, from other data transmitted via the same signaling mechanism (DTMF). We've shown that even such a simple implementation has a worst-case accuracy of 98.3 percent (or 98.9 percent when correcting for certain failures) and an expected accuracy of 99.4 percent. Finally, we have discussed how this system could be improved and deployed for real-world use.

Sources

BLUE, MR. 1997. DTMF Encoding and Decoding In C. Phrack Magazine, 7(50).

http://phrack.org/issues/50/13.html

CHEN, CHIOUGUEY J. 1996. Modified Goertzel Algorithm in DTMF Detection Using the TMS320C80. http://www.ti.com/lit/an/spra066/spra066.pdf

CLARKSON, KYLE AND JONES, DOUGLAS L. 2004. Goertzel's Algorithm. http://cnx.org/contents/kw4ccwOo@5/Goertzels-Algorithm

DIGIUM. 2002. dsp.c. http://doxygen.asterisk.org/asterisk1.0/dsp_8c-source.html

FOX, MARGALIT. 8 February, 2013. John E. Karlin, Who Lead the Way to All-Digit Dialing, Dies at 94. New York Times.

GAST, MATTHEW S. 2001. T1: A Survival Guide. O'Reilly, Cambridge, MA.

GOERTZEL, G. 1958. An Algorithm for the Evaluation of Finite Trigonometric Series. The American Mathematical Monthly, 65(1), 34-35. http://www.jstor.org/stable/2310304

ITU-T Recommendation E.164. 2011. The International Public Telecommunications Numbering Plan. http://www.itu.int/rec/T-REC-E.164/en

ITU-T Recommendation Q.22. 1988. Frequencies to be used for in-band signaling. http://www.itu.int/rec/T-REC-Q.22/en

ITU-T Recommendation Q.23. 1988. Technical Features of Pushbutton Telephone Sets. http://www.itu.int/rec/T-REC-Q.23/en

ITU-T Recommendation Q.24. 1988. Multifrequency push-button signal reception. http://www.itu.int/rec/T-REC-Q.24/en

LENHART, AMANDA. 2 September, 2010. Cell Phones and American adults. http://www.pewinternet.org/2010/09/02/cell-phones-and-american-adults/

MOCK, PAT. 1989. Add DTMF Generation and Decoding to DSP-mP Designs. http://www.ti.com/lit/an/spra168/spra168.pdf

NORTH AMERICAN NUMBERING PLAN ASSOCIATION. (N.D.). NANPA: Numbering Resources – NPA (Area) Codes. https://www.nationalnanpa.com/area_codes/index.html

ORLOWSKI, ANDREW. 30 January 2013. “The Death of Voice: Mobile phone calls now 50 per cent shorter”. The Register. http://www.theregister.co.uk/2013/01/30/mobile_phone_calls_shorter/

SAHIDULLAH, MD AND SAHA, GOUTAM. Comparison of Speech Activity Detection Techniques for Speaker Recognition. https://arxiv.org/pdf/1210.0297.pdf

SCHENKER, L. 1960. Pushbutton Calling with a Two-Group Voice-Frequency Code. The Bell System technical Journal, 39(1).

ZAPATA COMPUTER TELEPHONY TECHNOLOGY. 2001. goertzel.c.

https://sourcecodebrowser.com/zapata/1.0.1/goertzel_8c_source.html