Automatic Speech Recognition For Dysarthria: A Survey

Hemanshu Bhargav
Department of Industrial Engineering
Toronto Metropolitan University
Toronto, Canada
hbhargav@ryerson.ca 0000-0002-6412-6521

Abstract— The Time-Sensitive Telephony Problem affects smartphone users with dysarthric speech and requires a solution using Automatic Speech Recognition (ASR). ASR refers to the process by which a computer detects and transcribes speech [9, 81]. To the author's knowledge, no literature review of ASR has been conducted in the context of the Time-Sensitive Telephony Problem affecting smartphone users with dysarthric speech. A review of the literature on the impact of Deep Neural Networks (DNNs) on the ASR pipeline, notable DNN variants, End to End speech recognition, characteristics of dysarthric speech, the current state of ASR systems developed for dysarthria, and ASR variants effective for dysarthric speech are presented. A lack of research expertise and study participants limits the volume of audio data which has been collected for dysarthric speech, and consequently prevents the establishment of a large training vocabulary and makes most ASR systems, including those designed for dysarthric speech, ineffective. Further, the research suggests that the inherent variability of dysarthric speech makes employing continuous speech recognition infeasible, even with recent advances in pretrained, hybrid End to End ASR transformers. The paper concludes with the recommendation of using a hybrid Attention-CTC transformer on a limited vocabulary, two-stage Wake-Word-Detection (WWD) model using Voice-Activity-Detection (VAD) optimized for smartphones.

Keywords— ASR, dysarthria, telephony, WWD, KWS, Attention, CTC

I. Introduction

automatically detecting speech voiced from a human speaker and producing textual transcriptions of the continuous utterances [9]. Reference [81] defines ASR as "process of converting a speech signal to a sequence of words, by means of an algorithm implemented as a computer program."

Like any sequence-to-sequence modelling problem, ASR requires that features be extracted from sequences of input vectors (audio data) from which sequences of output vectors (transcriptions of the spoken audio data) can be produced [25, 28]. The ASR pipeline, however, adds a few layers of complexity, as the encoding and decoding of the sequence-to-sequence architecture must recognize and account for errors in speech, since ASR systems usually are developed for Speech-to-Text (STT) or Text-to-Speech (TTS) applications. Speech, like other Natural Language Processing (NLP) tasks, requires a deep understanding of *context* [48]. By understanding context, predictions can be made in the absence of noiseless, uncorrupted audio streams that differ in pronunciation, accentuation and a plethora of other variables which have been modeled through decades of research.

As shown in Fig. 1, before DNNs were applied to speech, the ASR pipeline began with a language model, which was used to construct a pronunciation model, which in-turn created a Gaussian Mixture Model (GMM) based acoustic model, and finally produced output (denoted as "Y" using features extracted from frames of audio data [13]. However, due to recent improvements in computing technology, and the result of improvements in the backpropagation algorithm, each of the above noted components has been replaced with neural-network-based counterparts, which has resulted in a complete DNN-based pipeline called "End-to-End Speech Recognition" [13].

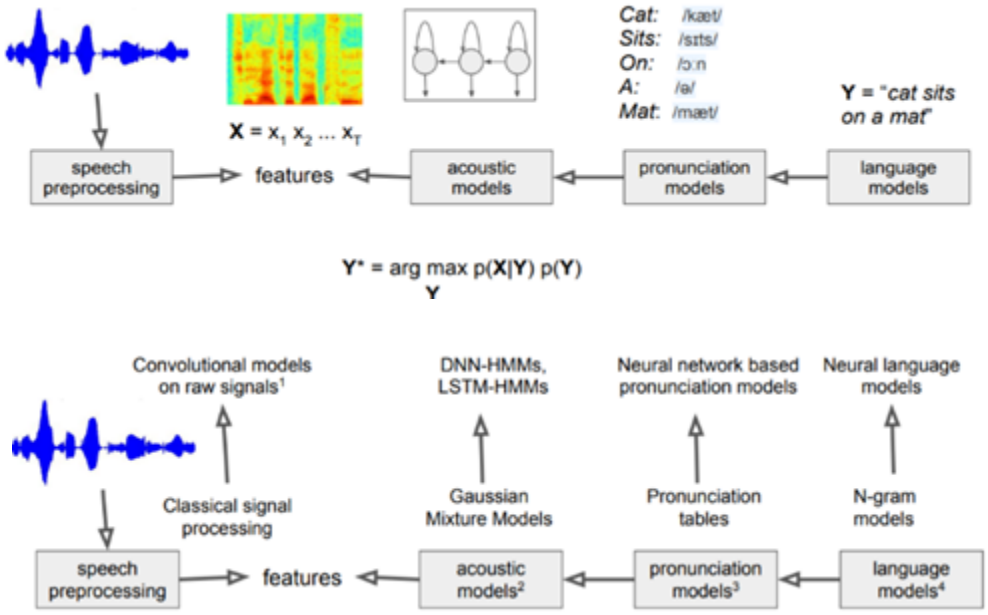

Fig 1. Evolution of Speech Recognition Process. Source: From Reference [13]

A. Feature Extraction

Before speech recognition can take place, speech data (usually presented as speech signals) must be preprocessed such that it is discretized in segments which are ready for training or classification. The most common approach is using MFCCs, and requires five or six steps [90, 103]. To convert raw audio data (e.g. a wave file) into trainable features, first and foremost, the audio data is segmented into frames based on time, usually ranging from 20-30ms, which can be done using Short Time Fourier Transform [89, 103]. Next, a Hamming function is applied to create overlapping sliding windows (similar to the convolution operator in Convolutional Neural Networks (CNNs)) on the frames of data so that resultant segments are continuous and free of distortion [103]. Third, Fast Fourier Transformations are applied to the time-based sliding windows, thereby producing an audio spectrum of frequencies [90, 103]. The Mel scale is a useful conversion as it "is a nonlinear scale in all frequency bands following the sensitivities of human ear when hearing sounds." [90, p.4]

Applying a linear space operation produces triangular filters (Mel Filters), from which the energy per frame per window can be calculated to filter frequency bands using the triangular filters produced, creating a Mel Spectrogram [90]. Then a logarithmic scale can be applied to create a Log Mel Spectrogram [90, 103]. Finally, taking the discrete cosine transformation of the Mel Spectrogram or Log Mel Spectrogram results in the MFCCs, which can be inputted

into an appropriate speech recognition algorithm [90, 103]. The steps are illustrated in Fig. 2. Although plentiful, applications of MFCC in other fields of speech recognition (e.g., speaker identification) are out of the scope of this paper.

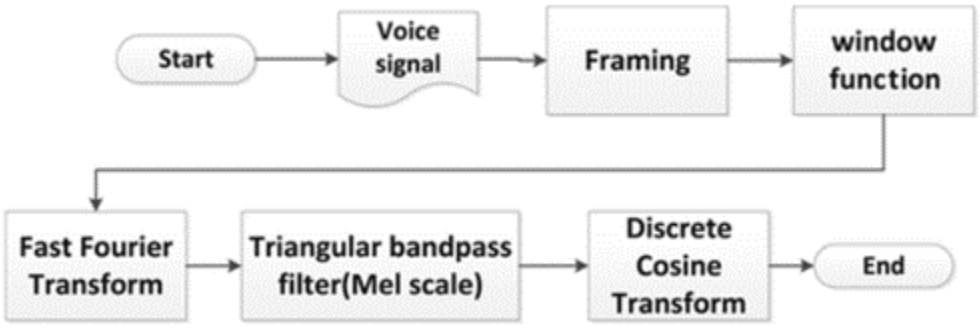

Fig. 2. Feature Extraction Using MFCC. Source: From Reference [90]

B. Performance

Throughout this paper, several DNN architectures and advances in speech recognition are explored. One recurring theme which provides evidence for a technique or algorithm providing advancement to the field is the measurement of performance. In the field of ASR, performance can be measured in a number of methods, but one such technique is both universal and can be considered mandatory: Word-Error-Rate (WER). In addition to WER, performance can also be measured through improvements in access time (for interface-based advances), better detection of segments or features of processed speech, reduction of training time, Character-Error-Rate (CER) and Label-Error-Rate (LER) [25-26].

C. Omissions

Since 2006, deep learning algorithms have overtaken classical, Hidden-Markov-Model (HMM) based speech recognition systems [72]. Although HMMs are still used to complement DNNs, namely as DNN-HMM systems, recent advances in End-to-End speech recognition techniques have made purely classical ASR systems obsolete. However, HMMs do form the basis of many popular DNN-HMM systems, as well as providing the foundation for the theoretical underpinnings of End-to-End speech recognition systems, as the modern objective functions are based on the HMM forward-backward algorithm, and the Viberti algorithm remains in use [25, 104]. Finally, because [33] established that phonemes are less relevant to ASR for dysarthric (see Section IV), linguistic features of ASR are not discussed.

II. Deep Neural Networks (DNN) & The ASR Pipeline

A neural network is a probabilistic network of neurons (a combination of weights and biases) which can classify patterns through function approximation of input data. A neural network is considered to be deep if it is composed of several hidden layers [14].

Although neural networks have been used with HMMs since the 1990s, it was in 2006 when each component of the ASR pipeline was outperformed using DNNs (in most cases, DNN-HMM models) [98]. In today's time, DNNs have replaced classical ASR completely. While End-to-End Speech Recognition is now the norm, DNNs began outperforming HMM-GMM based speech recognition models a decade prior. As alluded to earlier, this transformation from classical speech recognition systems to End-to-End speech recognition systems occurred through the replacement of each of the four steps of the ASR pipeline, namely: speech preprocessing, acoustic modelling, pronunciation modelling and language modelling [12].

A. Restricted Boltzmann Machines & Speech Preprocessing

The first major breakthrough in DNN-based speech recognition for speech preprocessing took place in 2011, when Jaitley and Hinton presented evidence that Restricted Boltzmann Machines (RBMs), what many considered to be the one of the first modern implementations of DNNs, when trained on "statistics of raw [audio] signals", regularly outperformed classical techniques [15, p. 5884]. Specifically, the paper showed that RBMs better learned the features which are relevant to speech recognition "than the traditional features such as mel filter banks" [15, p. 5884]. Further, their experiments found that the "detected features can be used to achieve better performance in phoneme recognition on the TIMIT corpus than most of the state-of-the-art speaker-independent systems built on Mel filter banks and MFCCs" [15, p. 5884]. This breakthrough indicates to researchers that DNNs (or RBMs at the time) are capable of analyzing patterns in speech which decades of statistical analysis theory cannot.

B. Deep Neural Networks & Acoustic Modelling

The second breakthrough in DNN-based speech recognition was not a single experiment, but the insight on a compilation of experiments which found that DNNs did indeed outperform GMMS in speech modelling by a significantly large margin [16]. The research groups (led by Hinton) attempted to feed "several frames of coefficients as input and produced posterior probabilities over HMM states as output" using feed-forward neural networks [16, p. 82]. Note that these promising results were found using only feed-forward neural networks, and not Recurrent Neural Networks (RNNs) and it's more advanced varieties which are much more suitable for speech recognition. Reference [55], a research group supervised by Hinton, again saw deep belief networks (equivalent to DNNs) used to replace classical acoustic models.

In the classical approach, researchers use the "temporal variability of speech and Gaussian mixture models (GMMs) to determine how well each state of each HMM fits a frame or a short window of frames of coefficients that represents the acoustic input." [16, p. 5884] A feed-forward neural network however, "takes several frames of coefficients as input and produces posterior probabilities over HMM states as output." [16, p. 5884]

Hinton's seminal paper did more than illustrate years of DNN's improvements over GMM acoustic modelling, it also served as a signal to researchers that DNNs were capable of changing what was possible with computer science-- opening a gateway for a wide range of applications. However, Hinton made one important clarification, this plethora of possibilities was possible not because of advances in algorithmic research (although Hinton did

advance DNN research significantly through his backpropagation algorithm), but because advances in hardware technology made it possible for GPUs to train copious amounts of data [16]. As a result, the popularization of DNNs, and the resultant gains found in the numerous studies establishing strong empirical gains observed in replacing GMMs with DNNs can be attributed to both advances in highly specialized algorithms (backpropagation), as well as the accessibility of powerful GPUs which can train large amounts of data in mere seconds [16].

C. Long Short Term Memory & Language Modelling

The third breakthrough took place in 2015, when RNNs, specifically Long Short Term Memory (LSTM) networks, were used to create pronunciation tables which could successfully replace classical pronunciation modelling [17]. Although RNNs expanded the functionality of feed-forward neural networks by allowing neural networks to store information, a task necessary for speech recognition due to the nature of language's interdependencies (since language cannot be understood without context), they suffer from a particular problem: long-term dependencies [18]. The solution proposed by Hochreiter and Schmidhuber [18] prevented the problem of long-term dependencies by incorporating additional functionality, or gates which can store predicted vector data for longer durations.

D. RNNs & Pronunciation Modelling

Finally, Rao, Peng, Sak and Beaufays [17] furthered speech recognition through their DNN-based pronunciation modelling. The final breakthrough (chronologically the first) in DNN-based ASR occurred when Mikolov and Karafiat used RNNs to construct neural language models which reduced WER on speech recognition from the Wall Street Journal (WSJ) dataset [19].

E. Beyond LSTM: GRU & Li-GRU

Since Rao, Peng, Sak and Beaufays' paper, there have been a number of developments in ASR using another type of RNN: the Gated Recurrent Unit (GRU). Similar to the LSTM architecture, GRU architectures solve the problem of long-term dependencies that traditional RNNs face, but do so by simplifying the operation [20, 21, 22].

First introduced by reference [20], the Gated Recurrent Unit (GRU) bears many similarities to the LSTM architecture, as it attempts to solve the long-term dependency problem faced by RNNs in empirical testing of time-series or textual data which carry forward dependencies across multiple layers.

Where the GRU differs is in complexity. The GRU simplifies many of the operations which LSTM proposes by merging functionality. As shown in Fig. 3, by merging the cell state and hidden state, which is accomplished by merging the forget neural network, or simply the forget gate with the input gate (both found in general LSTM architectures), a single gate accomplishes the same objective, but removes the plus junction operator, so element-wise addition between input data vectors is no longer required. By doing so, the GRU reduces the complexity of operations, since each layer of input vector data has fewer operations performed.

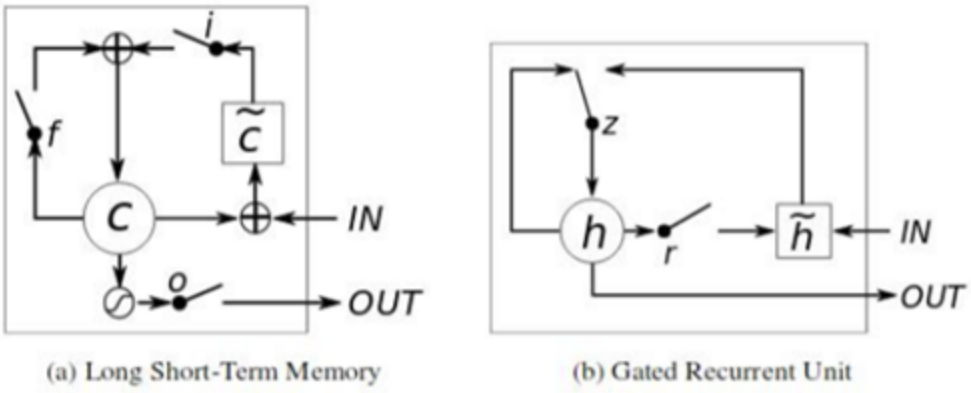

Fig. 3: Comparison of LSTM and GRU Hidden Units. Source: From Reference [21]

In GRU architectures, the input gate and forget gate are both merged into one singular gate: the update gate. With the input and forget gates merged into a new update gate, z, or zt for update at an instance of time (or adaptive constant), and the output gate is replaced with a reset gate, r, or rt for reset at an instance of time. [22] The update gate indicates which of the vector data values are kept, and the reset gate resets the cell/hidden state to the default state of having no memory (which of the vector data values are forgotten). This allows the update and reset gates to collect and retain information in one step, and again, the number of total operations is reduced [22]. Mathematically, the update gate is equal to the sigmoid applied on the weight of the update gate, multiplied by the raw vector input and summed with the weight of the candidate activation.

$$z_t^j = \sigma\left(W_z x_t + U_z h_{t-1}\right)^j \quad (1)$$

$$r_t^j = \sigma\left(W_r x_t + U_r h_{t-1}\right)^j \quad (2)$$

Likewise, the reset gate is an element-wise multiplication of raw input vector data with the weight of the reset gate matrix, summed with weighted (U) candidate activation, and "effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to *forget* the previously computed state" [21, p.4].

A key difference here with the LSTM is that the entire state of the architecture is updated and reset (thereby simplifying operations). Consequently, the LSTM can control how much new data passes through the forget gate, whereas the GRU cannot, but can control how the candidate activation function adds to or updates the update gate [21]. In summary, the LSTM has three gates: input, forget, and output; whereas the GRU only has two: update and reset.

The Light Gated Recurrent Unit (Li-GRU) takes simplification one step further, in which the reset gate is removed and ReLU activations are used" [24, p.5]. First proposed by [23, 24, p. 5], the Li-GRU is a DNN architecture specifically designed to reduce training time by 30%, "but to slightly improve the recognition performance" as well. Reference [77, p. 2955] confirms the performance results of the Li-GRU, creating another variation, the "light Bayesian recurrent unit (Li-BRU)."

Put succinctly, the state and memory differences of the three architectures summarized in Table I:

TABLE I. Table I Comparison of RNN variants

LSTM	GRU	Li-GRU
● Long-term memory is stored in the cell state (ct) ● Short-term memory is stored in the hidden state (ht)	● Both long-term and short-term memory are combined into a singular candidate activation (ht). ● I/O and forget are replaced w/ update and reset	● Changes activation function from tanh to ReLU ● Removes the reset gate, leaving only a singular update gate

Although the presented variants of DNNs had outperformed classical speech recognition techniques well before End-to-End speech recognition techniques took over, this did not make classical speech recognition obsolete. Even today, there exist implementations of DNN-HMM models, architectures which combine DNNs and HMMs, however, these do not perform as well as current End-to-End approaches [87]. As [98, p.2504] states, DNNs for ASR can be implemented using two techniques, one where the DNN acts "as [a] feature extractor for a standard GMM-based HMM system", or two, the hybrid approach in which the DNN "uses the neural network to compute HMM/GMM state posteriors." [98] Reference [99] confirms this approach for both ASR and Keyword Spotting (KWS, see Section V. A).

As noted by [16], when DNNs first began replacing HMMs, they did not do so in entirety. Instead, the revolution of DNNs arguably began in 2011 when DNNs were used in conjunction with HMMs, with DNNs creating "posterior probabilities over HMM states as output." [16, p.82] Today, however, HMMs are no longer required to be used in conjunction with DNNs. Instead, the entire ASR pipeline can be implemented through variants of DNNs alone, in End to End architectures. Reference [98] reports several KWS (and ASR) publications which show DNN-HMM models outperformed, including one where an LSTM-CTC model outperformed DNN-HMM on the WSJ0 dataset by a 9.8 figure-of-merit (as established KWS evaluation metric) point lead.

III. End to End Speech Recognition

A. Neural Machine Translation & Sequence-to-Sequence Architectures

Reference [20] introduced the concept of the GRU, and it did so motivated by the main objective of their publication: proposing the novel concept of the encoder-decoder architecture. The framework proposed by [20, p. 1]

“consists of two recurrent neural networks (RNN) that act as an encoder and a decoder pair. The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence.”

This encoder-decoder architecture was used to apply neural networks towards Statistical Machine Translation (SMT), by converting source phrases into target phrases [20]. SMT is the process of using statistical analysis to translate one language (i.e. the source sequence) to another language (i.e. the target sequence) [81]. Although SMT was created for machine translation (language translation performed by a computer), it is also used to translate audio into text belonging to the same language [81, 82].

Shortly after the proposition of the encoder-decoder architecture, [83] proposed the concept of Neural Machine Translation (NMT): the process of using neural networks, specifically encoder-decoder pairs, in place of statistical techniques, for machine translation. Soon after, [84, p.2] proposed the concept of sequence-to-sequence models, an architecture which was “closely related” to the NMT technique proposed by [83], “who were the first to map the entire input sentence to vector.” Reference [85, p. 2], a team consisting of many of the same authors from [20], confirm the role of RNN encoder-decoder pairs in NMT, while introducing a “gated recursive convolutional recurrent neural network (grConv)”, building from their previous work in [20], but also acknowledging the work of [83] and [84].

To sum up, encoder-decoder pairs of RNNs were used to learn and translate phrase representations, which led to the formation of NMT and finally Sequence to Sequence architectures, a concept very similar to NMT, but not limited solely to language translation [20, 83, 84].

In the same year, [78, p. 1] proposed the attention framework, a model which replaces the RNN components in Sequence to Sequence and NMT models. The attention model builds upon the concepts of RNN encoder-decoder pairs, but unlike the encoder-decoder pair, the attention mechanism “learns to align and translate simultaneously.” As [78, p. 3] states, the attention mechanism uses a “bidirectional RNN as an encoder … and a decoder that emulates searching through a source sentence during decoding a translation.” Reference [78, p. 9] reports that the previous NMT approach of using “fixed-length context vectors is problematic for translating long sentences.” The attention mechanism however, solves this problem by “letting a model (soft-)search for a set of input words, or their annotations computed by an encoder, when generating each target word. This frees the model from having to encode a whole source sentence into a fixed-length vector, and also lets the model focus only on information relevant to the generation of the next target word.” [78, p. 9]

B. Connectionist Temporal Classification (CTC)

In ASR, oftentimes the audio data contains more information than the corresponding transcriptions, as transcriptions often omit pauses (silence) and filler words, such as “um”, “ok” and so on. A similar problem was alluded to earlier with NMT, which was solved using the attention mechanism (incepted after Connectionist Temporal Classification). In both cases, the key issue is alignment: a one-to-one mapping between source and target sequence cannot be formed.

This presents a conundrum, as when classification with DNNs is performed, the total size of the output vectors (target, i.e., transcriptions) may be smaller than the total size of the input vectors (audio data). Or as [25] denotes, if the input is $X = \{x1, \ldots, xT\}$, and the output is $Z = \{z1, \ldots zU\}$, then $U<=T$. CTC is an algorithm which can

overcome this problem, using its temporal classifier, which produces probabilities (denoted as Y) over T dimension/size of vector data [25].

Connectionist Temporal Classification (CTC) takes the task of labelling unsegmented data sequences (over a dataset S) by connecting them using RNNs [25]. To do so, one must first perform framewise classification, the process of labelling each frame in a sequence of audio data, and then train the classifier to produce "alignments" over all probabilities Y [25]. Then, to remove the filler words (called "blanks") (denoted as L) (3) is used, and to ensure that the probabilities of the output/target do not exceed the probabilities of the input, (4) is used [25]. To remove the blanks and (and all duplicate labels) a many-to-one mapping is used, and the RNN decoder uses "best path decoding" [25, p3]. The result is "is an objective function that allows an RNN to be trained for sequence transcription tasks without requiring any prior alignment between the input and target sequence." [86, p. 2]

$$p(l\,|x\,) = \sum_{\pi \in \beta^{-1}} p(\pi|x) \quad (3)$$

$$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^{t} \quad ,\forall \pi \ \in L'^{T} \quad (4)$$

As usual, edit distance serves as the evaluation/error metric label error rate [25]. As [25, p. 372] describes, CTC is implemented using dynamic programming, and rather than be trained on the alignment of I/O sequences, all possible alignments are traversed in a manner "similar to the forward-backward algorithm for HMMs", formalized by Rabiner [104].

C. Attention

In ordinary sequence-to-sequence architectures, an encoder neural network encodes input vector data (such as words or audio frames), which are processed into hidden states, and then inputted into the decoder neural network, which outputs predictions (such as the next word in sequence). In such an architecture, LSTM and GRU networks are used to solve the long-term dependency problem faced by traditional RNNs, as entire vectors of interdependent information are encoded and decoded across time [27,76].

In an attention-based neural network however, each atomic unit of data, such as a word or MFCC feature is fed directly into the encoder, then the hidden state and finally the decoder, presenting an alternative to CTC [27, 28, 29]. Rather than process entire batches of interdependent information, attention-based networks prevent the issue of long-term dependencies by propagating (or transforming) the data through the entire pipeline, so dependencies are eliminated entirely [27]. As seen in Fig. 4, this process is done concurrently with the input and output pipeline, rather than using the consecutive I/O strategy used by past sequence-to-sequence models.

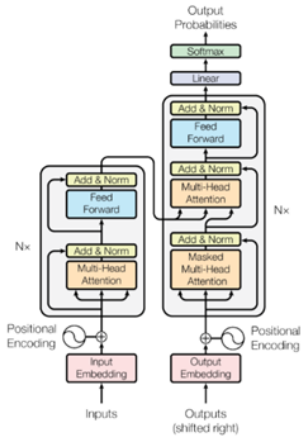

Fig. 4. From Reference [27] Joint Training Using the Attention Transformer

The attention mechanism was first proposed by [78] as an alternative to CTC and RNN-based architectures. Reference [78] proposed using an RNN to encode source sequences, while simultaneously learning alignments using the attention mechanism, [27] extended the attention mechanism one step further, by replacing the RNN with attention as well, in their aptly titled publication, *Attention is All You Need*. Reference [79] confirmed that the attention mechanism can replace RNNs or CNNs in NLP tasks. This new mechanism proposed by [78], commonly referred to as a transformer, is capable of transforming an entire source sequence into its target sequence, rather than encoding and decoding consecutively fed segments. Since then, transformers have been applied to a plethora of machine learning tasks, including the development of new pretraining/fine-tuning models such as GPT and BERT [87, 88].

One particularly useful transformer for speech recognition is by [76], which is similar to its NLP predecessor, proposed by [80]. The transformer in [76, p. 1] “Listen Attend and Spell” originally described processing log mel spectrogram vectors through two-layer CNNs, encoded through LSTMs, but since then other architectures have been used as well [27, 76, 26].

One particularly relevant application of the Listen, Attend and Spell architecture is in speech preprocessing using Specaugment [75]. Specaugments uses Mel spectrograms, which perform data augmentation, the process of reducing the size ASR's input data (i.e., log Mel spectrogram) and extrapolates the information (thereby producing more samples from one given sample) for better performance (similar to dimensionality reduction) but done by capitalizing on a more modern technique: the attention transformer [75]. Previously, such tasks were conducted using HMMs or time delayed DNNs [105-106].

D. Hybrid CTC-Attention Architecture

Although the attention-based mechanism was introduced as a potential replacement to the CTC architecture, several research groups have shown that hybrid approaches which utilize select segments of both techniques provide significant performance gains in various areas of speech recognition.

The hybrid CTC-Attention mechanism benefits from CTC's effective objective function which mitigates the alignment problem of source and target sequences, as well as from the Attention mechanism's ability to simultaneously learn and train alignments [57]. The result is "a multi objective learning framework" for the encoding and "combining both attention-based and CTC scores in a one-pass beam search algorithm to further eliminate irregular alignments." [57, p. 1240] Further, [84, p.2] suggests a potential weakness of CTC, as it "assumes a monotonic alignment between the inputs and the outputs."

Specifically, the hybrid CTC-Attention architecture solves several problems which traditional architectures cannot, namely: "stepwise refinement", "linguistic information", "conditional independent assumptions", "complex decoding" and "incoherence in optimization" [57, p. 1240]. Reference [57, p. 1240] also claims that as a consequence of these problems, "it is quite difficult for nonexperts to use/develop ASR systems for new applications, especially for new languages."

Following its introduction, the hybrid architecture was used to improve ASR capabilities in identifying pronunciation errors, chewing or swallowing noise, recognizing accentuation, generating audio adversarial examples audio-visual recognition and for improving performance in the presence of noise [58-62, 65]. In a context more relevant to this paper, Inaguma and Kawahara [63] proposed that the hybrid approach can even replace Voice Activity Detection (VAD) by using the same probabilities proposed by CTC to decide whether the ASR system has detected speech or not. Reference [64] combines the hybrid approach with pretraining on the encoder using Facebook's high-performing wav2vec 2.0 on acoustic and language models.

Reference [69] applies the hybrid approach to online ASR systems to simplify the "non-trivial problem" of deployment by streaming data rather than requiring offline loading of data. Reference [69, p. 1452] cites that the online application of the hybrid architecture "exhibit[s] comparable performance" to DNN/HMM architectures. Reference [66] takes the hybrid approach one step further, by limiting the number of audio frames the attention transformer operates on. Reference [67] adds CNNs to the mix, and [68] used the vanilla hybrid approach to achieve the best results of ASR on the LibriSpeech dataset up until 2018.

This well-documented hybrid CTC-Attention mechanism has garnered such attention that researchers and donors have collaborated to make a PyTorch-based implementation freely and publicly available on Github [24]. Speechbrain is an all-in-one speech toolkit which provides developers an extensible template of the hybrid mechanism as well as allowing them to customize their choice of encoder and decoder leveraging community-supported libraries [24].

E. Wav2Vec 2.0

As alluded to earlier, transformers have been applied to many NLP tasks, including pre-training and finetuning. Wav2vec 2.0 is one such revised, self-supervised pretraining model which learns patterns in unlabelled audio data [51]. First, audio signals are compressed into latent space representations using an acoustic model encoder (five-layer CNN) [51]. Then the latent space vectors are processed into the decoder which "combines multiple time-steps of the encoder to obtain contextualized representations", or outputs [51]. The decoder also uses CNNs, this time using neural language models and applying beam search on four-gram-based, character-based, and word-based [51]. Finally, as [51, p. 2] explains, "Both networks are then used to compute the objective function." The result is a model which has been trained to learn unlabelled speech data, so that it can be finetuned by researchers for their own labelled speech recognition tasks.

IV. Dysarthria & ASR

A. Factors Affecting Dysarthric Speech

As [70] proves, dysarthric speech is not affected by intelligibility in a semantic context, as the dysarthric speaker is sound in their intent of speech, but that articulatory factors reduce a human or computer's ability to distinguish between pronounced vowels. This spatial deviation is evident in comparisons of waveform analysis as seen in Fig. 5. As indicated by [70, p. 1], such speech is also subject to slurring and "lack of articulatory control [which] can lead to various involuntary non-speech sounds."

Knowing that dysarthric speech is highly variable is not enough to develop an effective speech recognizer, one must also comprehend which factors of speech recognition influence ASR accuracy of dysarthric speakers. [56, p. 1] states, "dysarthric speech is often associated with reduced vocal tract volume and tongue flexibility, atypical speech prosody, imprecise articulation, and variable speech rate - factors that all reduce speech intelligibility."

As of 2015, [54, p. 3924] found 21 publications (including those which are based on the datasets used in this paper) which explored the influence of factors which affect all ASR systems, or "general factors", and those which "are specific to dysarthric speech", or "specific factors". According to the results of [54, p. 3924], general factors are "speech mode, speaker mode, vocabulary size and speaking styles", whereas specific factors are "speech intelligibility, severity and intra-speaker variability". Speaker mode consists of one of three types, each differentiated by the dependence of the ASR system on the speaker: speaker dependent (SD), speaker independent (SI) and speaker adaptation (SA) [54]. SD systems are those in which the user is utilized in the ASR systems's training [54]. A SI system is one in which the user is not involved in training, and a SA system is one where the "ASR system can adapt to the user gradually during use." [54, p. 3925]. While general factors had "little influence" on ASR accuracy for dysarthric speech, specific factors, particularly intelligibility and severity had "significant influence" on dysarthric speech [54]. As such, although severity and intelligibility strongly influenced ASR accuracy, [54, p. 3931] found that SD systems are more accurate than SA for "severe dysarthric speech."

Reference [71, p. 4924], which was one of the publications studied by [54], concludes their findings that "speaker-independent (SI) speech recognition systems remain ill-suited to this population because of the considerable deviation of dysarthric speech from the assumed norm in these systems." Reference [71, p.4924] establishes that the WER for systems designed for dysarthric speech perform poorly compared to those for non-disordered speech, even for "small-vocabulary recognition tasks."

B. Existing ASR Systems Specialized for Dysarthric Speech

One of the earliest ASR systems for dysarthric speech was developed by the researchers of the UA Speech database themselves, who collected the largest dataset of dysarthric speech to date (discussed further in DATA COLLECTION), complete with articulatory and video analysis [74]. However, because the system was designed as a LVCSR, it is suspect to the variability issues of dysarthric speech which are detrimental for telephony functions, and does not use modern DNN technology [31,32]. Using the datasets publicly available, the STARDUST and homeService projects developed ASR system specifically for dysarthric speech, but the STARDUST project does not use DNNs, and the homeService project only uses two datasets for dysarthric speech [33, 34, 35].

The STARDUST project recruited eight dysarthric speakers voicing 30 samples of 10-12 words per speaker [33]. A key finding of the study was that ASR systems trained specifically on dysarthric speech recognizers do not suffer problems faced normally during machine translation, and that both speech intelligibility and phonemes are less relevant to ASR for dysarthric speakers [33]. Further, the study provided concrete evidence that even in limited vocabulary systems, words which are similar in pronunciation, specifically, “on” and “off”, are frequently misrecognized [33, p. 1]. Replacing just one of these similar sounding words with an alternative, such as “standby”, can improve both system and user performance [33]. Although the STARDUST system succeeds in utilizing a SD limited vocabulary system to mitigate the effects of fatigue on dysarthric speakers, the system does not perform as well as systems designed for non-disordered speech [33, 71]. This is largely due to the release date of the project, 2006, a time where classical speech recognition systems were still in use, and End-to-End speech recognition systems had not been conceived [33, 72]. The homeService project poses a similar solution to the STARDUST project, by utilizing a limited vocabulary system limited to a 30 word vocabulary, which is also SD and but adds not only modern DL algorithms, but personalization in the form of a Personal Adaptive Listener (PAL) [35].

Reference [73] performed a study regarding three commercial, online speech recognition systems designed for general use (i.e., assuming non-disordered speech for the majority of users). The study used a single participant with dysarthric speech and one control, both of whom used two continuous speech recognition systems (i.e., Microsoft Dictation and Dragon Naturally Speaking) and one discrete speech recognition system (i.e., VoicePad Platinum) [73]. The *in situ* testing involved voicing 10 sentences and 30 noun phrases into each of the systems, for a total of 21 hours [73]. Their study concluded that continuous speech recognition is preferred for non-disordered speech, but in dysarthric speakers, discrete, single-word utterances “allows additional time for language processing and word retrieval.” [73, p. 193] As [71, p. 4924] concluded regarding the study by [73], “Commercial systems from Microsoft, Dragon, and VoicePad recognized approximately 85% of words uttered by a non-dysarthric speaker on average, but only between 51.87% and 64.68% of words spoken by a person with mild dysarthria.” Although this study was limited to only one participant (and one control), it provides evidence that discrete systems, those which are of limited vocabulary such as predefined phrases, are more effective at recognizing dysarthric speech, even when the system in question has not been trained or designed primarily for dysarthric speech.

As described in [36, 37, 38] a count of at least 100,000 observations is required for DL to be effective. While it is possible to configure open-sourced ASR toolkits such as Py-Kaldi, a very popular general speech recognition toolkit reconfigured using PyTorch, this approach requires the insurmountable task of building the ASR pipeline from scratch [39]. Other possibilities include creating a personalized speaker-dependent model such as one from Snowboy or Dragon Naturally Speaking, but these solutions are privatized and unaffordable [40, 41, 42].

However, recent state-of-the-art attempts have been made to improve online ASR systems for dysarthric speakers. one such attempt was Google's project relate (previously known as project euphonia) [43, 44]. In this project, Google researchers collected data from individual volunteers using google services and applied their state-of-the-art recognition rates. According to their latest publication, accuracy has improved significantly, but the issue of offline connectivity remains, and the system has relied on data collected from users [43, 44].

To mitigate the control issue with user submitted samples, of which some may contain dysarthric speech of varying levels of severity, as well as intentionally included data from other speech impediments, I have instead sourced all data collected from academic institutions, where the condition of dysarthria has been confirmed, and can be used to model a system which does not rely on online connectivity, works for telephony functions, does not require the storing of personal data on commercial servers and is not limited to Google’s commercial applications.

None of these cases are particularly suited to solve the telephony issues described in this paper: The STARDUST system does not use modern DNN algorithms, homeService is limited in training data, VoicePad Platinum is not designed for dysarthric speech, and Google’s Project Relate is not available offline, nor does any of the above

surrender control of telephony functions. Thus, the variability of dysarthric speech persists, so even the most accurate STT system will not allow single-switch users to access time-sensitive functions, such as those required in an organization's automated menu system. To overcome the time-sensitive challenges, the system must be very accurate and correspond a user's voiced input directly to a particular function. In fact, in this scenario, context is less important, as the system need not recognize any words other than those within the limited vocabulary. Using a limited vocabulary thereby reduces the potential number of false matches and drastically reduces searching time, since the vocabulary can be limited to just telephony commands, rather than the entire English language.

C. Dysarthric Speech Data

In the case of speech recognition, the requirement of 100,00 data points is even more imperative due to the various stages of the pipeline, and because no pretrained models are available for dysarthric speech, a sizable corpus is imperative. The lack of state-of-the-art models contributes to the significant performance gap observed between dysarthric and disordered speech.

As shown in Table I, the data collected is composed of dysarthric speech samples (spoken in English) compiled from the TORGO, UA Speech, Nemours, and homeService datasets-- of which some are publicly available, and others have been permitted for research purposes. The TORGO dataset contains recordings of words, sentences and vowel sounds (for articulatory research) [9]. The UA Speech dataset is the largest dysarthric speech dataset collected to date, containing up to 514 sentences and telephony digits (0-9) voiced three times per speaker [32, 33]. The Nemours dataset contains recordings of spoken sentences and paragraphs [45]. The homeService is a dataset consisting primarily of voice-commands used for assistive and automation technology, similar to the needs described in this paper [34, 35].

While several non-English dysarthric speech datasets exist such as the CUHK Cantonese dataset and the Dutch dataset, their relevance to a WWD recognizer for English-speaking users is limited [46]. However, due to the limited availability of telephony-specific databases consisting of dysarthric speech, and because the EasyCall database does consist of telephony commands spoken by dysarthric speakers, an exception was made for the Italian EasyCall database [47]. Waveform analysis revealed the audio waveforms of Italian-spoken telephony commands (particularly the digits 0-9), were similar to the English-spoken digits found in the TORGO, UASpeech and homeService datasets. Further, some specific phrases (i.e., "end") sound similar to English phrases (i.e. "terminate"), and can be used as alternative telephony commands.

To the author's knowledge, no other publicly available English dysarthric speech datasets exist. Datasets of audio data of non-disordered speech however, are plentiful. Table II describes just a few parameters of the most commonly used publicly available data. Due to the varying size (some observations exceed one hour in length), the number of observations is a poor comparison metric, but the difference in magnitude between dysarthric and non-disordered speech is apparent– just one of the commonly used non-disordered datasets greatly exceeds the length of the total collection of dysarthric speech data available. Reference [47] confirms similar findings in their research and provides further evidence that the doubling of data in the TED-LIUM resulted in a directly proportional performance improvement.

TABLE II. Table II Metadata of Dysarthric Speech Data

Dataset	Number of Speakers	Number of Observations	Vocabulary Size	Hours
TORGO	7 (8)[a]	2762	1573	15
UA Speech	15 (13)	~80,000	455-541	102.7
Nemours	11	814	-[b]	<3
homeService	5	1286	33	10
EasyCall	31 (24)	21, 386	37[c]	-

[a] Parenthesis denotes number of controls

[b] The Nemours dataset consists of recordings with sentences and commands, so vocabulary size is not comparable.

[c] The EasyCall dataset contains 37 voice commands, but also includes recordings from 67 sentences.

TABLE III. Table III Comparison of Publicly Available English Datasets Containing Non-Disordered Speech

Dataset	Number of Speakers	Vocabulary Size	Hours
LibriSpeech	2484	200,00	1000
Common Voice	75,879	220,000	2,015 – 2,036[a]
GigaSpeech	-	-	10,000- 33,000
WSJ	-	35,875	80
WSJ (UK)	92	64,000	-
People’s Speech	-	-	87,000

SPGIS Speech	50,000	100,000	5,000
TED-LIUM	2,000	160,000	1,000

[a]Discrepancy in hours recorded is due to hours not validated/transcribed

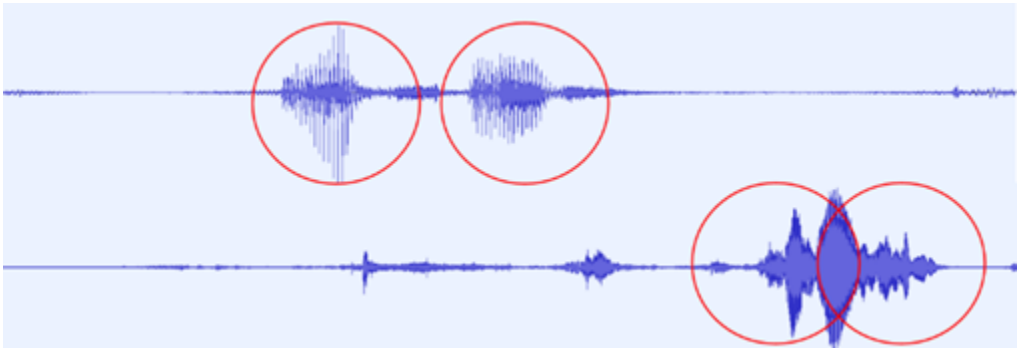

Fig. 5. Comparison of Two-Word Phrase Voiced By Participant & Control of UA Speech Dataset. Time taken to voice the actual phrase is much longer, and the distinction between words is not as evident as it is in the case of non-disordered speech.

V. ASR Variants Applicable to Dysarthria Telephony Issues

A. Keyword Spotting (KWS)

As observed by [36, 37, 38], an excess of 100,000 samples is required to create robust classifiers using neural networks. Thus, most speech recognizers are based on the concept of Large Vocabulary Continuous Speech Recognition (LVCSR), in which a recognizer is trained on large datasets and speech recognition is continuous so that real-time transcription is possible [30]. Keyword Spotting (KWS), sometimes referred to as Spoken Term Detection (STD), or Keyphrase Extraction, is a supervised subdivision of ASR (and NLP, when the data is textual) in which keywords (predefined words or phrases existing within the ASR corpus) are identified in speech signals [98]. Although [100] distinguishes STD as being able to use keywords which are not predefined by the system, whereas KWS does not, this paper does not make such a distinction between user-defined or system-recognized keywords. [101, 102] confirm that KWS has been a goal of general ASR since the 1990s, as it is necessary to detect domain-specific phrases in segments of speech. The method of search is generally the Dynamic Time Warping (DTW) algorithm [98, 100, 102]. As [98] and [102] confirm, because KWS is implemented with ASR systems, it usually exists as a LVCSR system. Reference [103, p.2] finds that LVCSR systems “are not suitable for a transfer learning targeting small or limited vocabulary” and in cases where the “input can be silence or contain background noise [an LVCSR system] be mistaken for speech and increases the false positive rate.”

Reference [98, p. 2506] establishes that although LSTMs (and other RNNs) have recently outperformed CNNs in KWS applications, CNNs "are better at modeling the local correlations in time and frequency in comparison to DNNs" and that "they better go along with changes between different speaking styles." Additionally, although [12] confirms the overall trend of RNNs outperform CNNs in ASR applications (due to the time dependent context of language),[107] shows one recent example (occurring after the publication of [98,107], and perhaps being out-of-scope for those publications) of a CNN outperforming RNNs in the case of WWD for non-disordered speech. This is a notable result, as [48] had established that WWD eliminates the need for context within language, requiring only a referential context. Also, recall that dysarthric speech is highly variable, so a solution capable of adapting to variability is promising, and that it may show commonalities of silence and diffuse behaviour (Fig. 5). These findings suggest CNNs (and their variants) may be appropriate for learning the variability of dysarthric speech in the context of WWD, since semantic context is less relevant in WWD. Fortunately, the hybrid CTC-Attention transformer proposed by [26] provides variants of CNNs for use in encoding. However, even if a transformer optimized with CNNs and DNNs is found, the Time-Sensitive-Telephony-Problem requires an architecture which can respond faster than LVCSR search algorithms allow.

As a result, since dysarthric speech often contains long periods of silence (see Fig. 5), KWS systems are prone to many of the same issues which affect users with dysarthria, but a limited vocabulary KWS can reduce the number of false matches and improve user experience.

B. Wake Word Detection (WWD)

In the case of voice-assistants, discrete, not continuous speech recognition is widely popular as the initial recognizer, due to Wake Up Word Detection (WUWD). WUWD, or simply Wake Word Detection (WWD), is the result of real-time, acoustic keyword spotting [48]. Unlike continuous speech recognition, this derivative of ASR utilizes a keyword, which switches the state of a recognizer from standby to listening [48]. The keyword, called a wake word, wakes the system up, so that the recognizer begins processing audio information only when the wake word is detected. The employment of this tactic allows the system to allocate significantly fewer resources to Voice Activity Detection (VAD) then complete LVCSR, and is necessary for privacy concerns, so that only the specific wake-word is recorded (temporarily), and all other speech is filtered into the "garbage" state of the HMM.

This real-time acoustic keyword search differs from the keyword search employed in continuous speech recognition. In continuous speech recognition, the system listens continuously for utterances of speech, and transcribes the real-time utterances as they are spoken, into text. As such, resource consumption is much higher, and the performance is also slower, because the entire training vocabulary must be searched to match the incoming audio data. This requires that the audio data (waveform) be analyzed, deconstructed into MFCC features, trained on the neural network and then matched with 100,00+ training observations. Acoustic keyword search however, only requires the matching of one keyword's waveform for the system to begin recognition.

As described by [48], user-defined keywords suffer from inaccurate recognition disproportionately than predefined keywords, as the user-defined keyword may be out-of-vocabulary (OOV). Introducing a keyword which has not been trained, or in the case of the collected data, represents a very small proportion, will not be enough for the recognizer to accurately spot the keyword. Even if the keyword is found within the lexicon vocabulary, a user-defined keyword requires the neural network to train the new utterance, which prevents the system's deployment in low-computing machines such as mobile devices.

Further, as shown earlier, dysarthric speech is highly variable. The utterance of a keyword spoken by an individual suffering from dysarthria in the morning may not be identical to the utterance produced in the evening, due to

fatigue and muscular dystrophy from the day [9]. As such, because user-defined keywords require more training, and because dysarthric speech is highly variable, it is necessary to establish predefined keywords. Such a task does not affect the usability of the system, as the system is intended to improve performance and reduce frustration solely of telephone commands, so the domain space of phrases is limited to telephone functions, which are predefined by the interface of the mobile application. As described in Bohouta's dissertation, the difference between general ASR and WWD lies in semantic context: WWD differentiates between alerting and referential contexts, whereas general ASR does not [48].

C. WWD Variants: Low Power, Two Stage & Personalized Models

As WWD is particularly useful in voice-assistant and smartphone applications, there is growing research in enabling WWD to be effective on low-resource devices (not to be confused with low-resource languages) [48, 97] .

Another potentially useful variant is the two stage WWD system. Two stage training has been applied to DNNs in contexts other than speech as well (see [95]), but is promising for the telephony issue of dysarthric ASR because it allows one WW for activating the system, and another WW (or set of WWs) for actual voice commands [96].

As shown in Section IV. A, the model described by [91] is an application of KWS, which is similar to the authors' commercial product Snowboy, a DNN-based WWD engine which allows users [91, 92]. Reference [93] also describes a novel personalized WWD using Support Vector Machines (SVM) and consequently better VAD performance.

Although personalized ASR models can be theoretically effective for dysarthric speech recognition due to the inherent variability and because they are SD systems, a speaker mode proven to be more accurate than SI systems [9, 54]. However, both the systems described by [91, 93] are both patented, require an online component, and are not trained on dysarthric speech, so their effectiveness in the application of the Time-Sensitive Telephony Problem is limited. Reference [94] employs a similar approach to [91], but also makes use of the more modern CTC objective function.

D. Voice Activity Detection (VAD)

When the theoretical framework is implemented, it requires the continuous use of the smartphone's microphone. As mentioned above, the WUW will prevent the execution of keyword voice commands made out-of-context, but background noise must be distinguished from when the user intends to voice the WUW and subsequently a keyword phrase. This requires a transition from the *standby* state to the *active* state. To accomplish this, Voice Activity Detection (VAD) is used. VAD is the process of identifying intelligible speech from background noise, usually based on whether the incoming audio passes a statistically generated threshold [49, 50]. This would require some initial testing, in which the user prompts the system to distinguish their voiced command from noise. Once complete, a threshold is identified for each individual (or set by the developer). As is in the case of ASR, modern VAD algorithms have now progressed to using DNNs for VAD, rather than statistical techniques. However, in the case of low-power mobile devices, and in the case of time-sensitive operations where VAD must occur almost instantaneously (since the time allotment to execute functions is restricted by service providers), statistical techniques hold relevance as a solution which can meet such strict performance demands. Although many statistical techniques exist, the frame energy based logistic regression classifier seems to provide better results than common techniques such as Zero-Crossing Rate or Short-Term Energy (or both) [49].

VI. Conclusion

This paper identifies the following patterns and commonalities among the ASR literature which are relevant to dysarthric speech applications. A complete timeline of the evolution of the ASR pipeline to its DNN counterparts is presented, leading up to the era of transformers. Evidence is provided that what began as a method for improving NMT and isolated ASR tasks ushered in new methodologies of speech recognition. This paper shows that transformers consisting of CTC-Attention hybrid mechanisms have replaced DNN-HMM hybrid systems, and the variants of DNNs which may be used conjunction with transformers to continually improve ASR performance are described in depth. The key characteristics of dysarthric speech, existing work towards ASR for dysarthric speech and the state of dysarthric speech datasets is explored. The research strongly suggests that hybrid Attention-CTC transformers outperform all other ASR techniques. Finally, the paper concludes by providing rationale that a two-stage WWD model (trained with a hybrid transformer), using VAD optimized for low-computing applications is an ideal solution to the Time-Sensitive Telephony Problem.

ACKNOWLEDGEMENTS

Bhargav thanks the Accessible Technology Program of Innovation, Science and Economic Development Canada for generously funding this research.

LIMITATIONS

There are a number of limitations with this literature review. First, the review is highly limited in scope and volume, as the review is not exhaustive, and only serves to guide readers which publications have made notable contributions to the application of ASR systems for dysarthric speech. Second, contributions which are deemed notable are highly biased to the author's own limited perception, ignorance and the lack of any research support from librarians or other experts normally involved in the process of conducting a literature review. Finally, the review is focused only on major developments in deep learning which have furthered the performance of ASR systems, and does not delve into improvements made using classical methods, due to their limited relevance in the age of big data and the specialized knowledge required to implement such systems.

REFERENCES

[9] F. Rudzicz, "Production knowledge in the recognition of dysarthric speech," Ph.D., University of Toronto (Canada), Canada, 2011. Accessed: Mar. 05, 2021. [Online]. Available: http://search.proquest.com/docview/920144730/abstract/A27C083F9BBA4210PQ/1

[10] H. H. Koester and S. Arthanat, "Text entry rate of access interfaces used by people with physical disabilities: A systematic review," *Assistive Technology*, vol. 30, no. 3, pp. 151–163, Sep. 2018, doi: 10.1080/10400435.2017.1291544.

[11] C. Havstam, M. Buchholz, and L. Hartelius, “Speech recognition and dysarthria: a single subject study of two individuals with profound impairment of speech and motor control,” Logopedics Phoniatrics Vocology, vol. 28, no. 2, pp. 81–90, Jan. 2003, doi: 10.1080/14015430310015372

[12] S. Alharbi *et al.*, “Automatic Speech Recognition: Systematic Literature Review,” *IEEE Access*, vol. 9, pp. 131858–131876, 2021, doi: 10.1109/ACCESS.2021.3112535.

[13] N. Jaitley, “cs224n-2017-lecture12.pdf,” *Natural Language Processing with Deep Learning CS224N/Ling284*, 2017. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lectures/cs224n-2017-lecture12.pdf (accessed Oct. 18, 2021).

[14] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[15] N. Jaitly and G. Hinton, “Learning a better representation of speech soundwaves using restricted boltzmann machines,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 5884–5887. doi: 10.1109/ICASSP.2011.5947700.

[16] G. Hinton *et al.*, “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012, doi: 10.1109/MSP.2012.2205597.

[17] K. Rao, F. Peng, H. Sak, and F. Beaufays, “Grapheme-to-phoneme conversion using Long Short-Term Memory recurrent neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 4225–4229. doi: 10.1109/ICASSP.2015.7178767.

[18] S. Hochreiter and J. Schmidhuber, “LSTM can Solve Hard Long Time Lag Problems,” in *Advances in Neural Information Processing Systems*, 1997, vol. 9. Accessed: Dec. 18, 2021. [Online]. Available: https://proceedings.neurips.cc/paper/1996/hash/a4d2f0d23dcc84ce983ff9157f8b7f88-Abstract.html

[19] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, *Recurrent neural network based language model*, vol. 2. 2010, p. 1048.

[20] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” *arXiv:1406.1078 [cs, stat]*, Sep. 2014, Accessed: Nov. 15, 2021. [Online]. Available: http://arxiv.org/abs/1406.1078

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *arXiv:1412.3555 [cs]*, Dec. 2014, Accessed: Nov. 15, 2021. [Online]. Available: http://arxiv.org/abs/1412.3555

[22] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Gated Feedback Recurrent Neural Networks,” *arXiv:1502.02367 [cs, stat]*, Jun. 2015, Accessed: Nov. 26, 2021. [Online]. Available: http://arxiv.org/abs/1502.02367

[23] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, “Light Gated Recurrent Units for Speech Recognition,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 2, pp. 92–102, Apr. 2018, doi: 10.1109/TETCI.2017.2762739.

[24]]M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, “Improving speech recognition by revising gated recurrent units,” arXiv:1710.00641 [cs], Sep. 2017, Accessed: Jan. 03, 2022. [Online]. Available: http://arxiv.org/abs/1710.00641

[25] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*, New York, NY, USA, Jun. 2006, pp. 369–376. doi: 10.1145/1143844.1143891.

[26] M. Ravanelli et al., “SpeechBrain: A General-Purpose Speech Toolkit,” arXiv:2106.04624 [cs, eess], Jun. 2021, Accessed: Jun. 30, 2021. [Online]. Available: http://arxiv.org/abs/2106.04624

[27] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 4960–4964. doi: 10.1109/ICASSP.2016.7472621.

[28] A. Vaswani *et al.*, “Attention Is All You Need,” Jun. 2017, Accessed: Dec. 18, 2021. [Online]. Available: https://arxiv.org/abs/1706.03762v5

[29] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 4945–4949. doi: 10.1109/ICASSP.2016.7472618.

[30] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 4960–4964. doi: 10.1109/ICASSP.2016.7472621.

[31] H. V. Sharma and M. Hasegawa-Johnson, “Acoustic model adaptation using in-domain background models for dysarthric speech recognition,” *Computer Speech & Language*, vol. 27, no. 6, pp. 1147–1162, Sep. 2013, doi: 10.1016/j.csl.2012.10.002.

[32] H. Kim *et al.*, “Dysarthric speech database for universal access research: INTERSPEECH 2008 - 9th Annual Conference of the International Speech Communication Association,” *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 1741–1744, Dec. 2008.

[33] M. Parker, S. Cunningham, P. Enderby, M. Hawley, and P. Green, “Automatic speech recognition and training for severely dysarthric users of assistive technology: The STARDUST project,” *Clinical Linguistics & Phonetics*, vol. 20, no. 2–3, pp. 149–156, Jan. 2006, doi: 10.1080/02699200400026884.

[34] M. Nicolao, S. Cunningham, P. Green, S. Deena, T. Hain, and H. Christensen, “Speech-enabled environmental control in an AAL setting for people with speech disorders: a case study,” in *IET International Conference on Technologies for Active and Assisted Living (TechAAL)*, London, UK, 2015, p. 6 .-6 . doi: 10.1049/ic.2015.0129.

[35] H. Christensen, I. Casanueva, S. Cunningham, P. Green, and T. Hain, “homeService: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition,” in *Proceedings of the Fourth Workshop on Speech and Language Processing for Assistive Technologies*, Grenoble, France, Aug. 2013, pp. 29–34. Accessed: Aug. 12, 2021. [Online]. Available: https://aclanthology.org/W13-3906

[36] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
140

[37] V. Sze, Y.-H. Chen, T.-J. Yang, and J. Emer, “Efficient Processing of Deep Neural Networks: A Tutorial and Survey,” pp. 1–32, 2017, [Online]. Available: http://arxiv.org/abs/1703.09039.

[38] J. Tang, C. Deng, and G.-B. Huang, “Extreme Learning Machine for Multilayer Perceptron,” IEEE Trans. Neural Networks Learn. Syst., vol. 27, no. 4, pp. 809–821, 2016, doi: 10.1109/TNNLS.2015.2424995.

[39] D. Can, V. R. Martinez, P. Papadopoulos, and S. S. Narayanan, “Pykaldi: A Python Wrapper for Kaldi,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5889–5893. doi: 10.1109/ICASSP.2018.8462463.

[40] G. Chen, C. Parada, and G. Heigold, “Small-footprint keyword spotting using deep neural networks,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 4087–4091. doi: 10.1109/ICASSP.2014.6854370.

[41] “Dragon Speech Recognition - Get More Done by Voice | Nuance,” *Nuance Communications*. https://www.nuance.com/dragon.html (accessed Dec. 18, 2021).

[42] G. Chen, C. Parada, and T. N. Sainath, “Query-by-example keyword spotting using long short-term memory networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 5236–5240. doi: 10.1109/ICASSP.2015.7178970.

[43] J. Green *et al.*, *Automatic Speech Recognition of Disordered Speech: Personalized Models Outperforming Human Listeners on Short Phrases*. 2021, p. 4782. doi: 10.21437/Interspeech.2021-1384.

[44] B. MacDonald *et al.*, “Disordered Speech Data Collection: Lessons Learned at 1 Million Utterances from Project Euphonia,” 2021.

[45] J. Polikoff and H. Bunnell, “The Nemours Database of Dysarthric Speech: A Perceptual Analysis,” Jul. 2003.

[46] S. Liu *et al.*, “Recent Progress in the CUHK Dysarthric Speech Recognition System,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2267–2281, 2021, doi: 10.1109/TASLP.2021.3091805.

[47] R. Turrisi *et al.*, “EasyCall corpus: a dysarthric speech dataset,” *arXiv:2104.02542 [cs]*, Apr. 2021, Accessed: Dec. 18, 2021. [Online]. Available: http://arxiv.org/abs/2104.02542

[48] G. M. Bohouta, “Improving Wake-Up-Word and General Speech Recognition Systems,” Ph.D., Florida Institute of Technology, United States -- Florida. Accessed: Nov. 01, 2021. [Online]. Available: http://www.proquest.com/docview/2468704152/abstract/E7E91356B8A74538PQ/1

[49] G. Meoni, L. Pilato, and L. Fanucci, “A low power Voice Activity Detector for portable applications,” in *2018 14th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, Jul. 2018, pp. 41–44. doi: 10.1109/PRIME.2018.8430328.

[50] J.-H. Chang, N. S. Kim, and S. K. Mitra, “Voice activity detection based on multiple statistical models,” *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 1965–1976, Jun. 2006, doi: 10.1109/TSP.2006.874403.

[51] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised Pre-training for Speech Recognition,” *arXiv:1904.05862 [cs]*, Sep. 2019, Accessed: Dec. 18, 2021. [Online]. Available: http://arxiv.org/abs/1904.05862

[52] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[53] C. Jose, Y. Mishchenko, T. Senechal, A. Shah, A. Escott, and S. Vitaladevuni, “Accurate Detection of Wake Word Start and End Using a CNN,” *arXiv:2008.03790 [cs, eess]*, Aug. 2020, Accessed: Nov. 15, 2021. [Online]. Available: http://arxiv.org/abs/2008.03790

[54] M. B. Mustafa, F. Rosdi, S. S. Salim, and M. U. Mughal, "Exploring the influence of general and specific factors on the recognition accuracy of an ASR system for dysarthric speaker," *Expert Systems with Applications*, vol. 42, no. 8, pp. 3924–3932, May 2015, doi: 10.1016/j.eswa.2015.01.033.

[55] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic Modeling Using Deep Belief Networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, Jan. 2012, doi: 10.1109/TASL.2011.2109382.

[56] N. P. Narendra and P. Alku, "Dysarthric speech classification from coded telephone speech using glottal features," *Speech Communication*, vol. 110, pp. 47–55, Jul. 2019, doi: 10.1016/j.specom.2019.04.003.

[57] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, Dec. 2017, doi: 10.1109/JSTSP.2017.2763455.

[58] L. Zhang *et al.*, "End-to-End Automatic Pronunciation Error Detection Based on Improved Hybrid CTC/Attention Architecture," *Sensors (Basel)*, vol. 20, no. 7, p. 1809, Mar. 2020, doi: 10.3390/s20071809.

[59] A. Nakamura, K. Ohta, T. Saito, H. Mineno, D. Ikeda, and M. Nishimura, "Automatic Detection of Chewing and Swallowing Using Hybrid CTC/Attention," in *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, Oct. 2020, pp. 810–812. doi: 10.1109/GCCE50665.2020.9292024.

[60] Q. Gao, H. Wu, Y. Sun, and Y. Duan, "An End-to-End Speech Accent Recognition Method Based on Hybrid CTC/Attention Transformer ASR," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 7253–7257. doi: 10.1109/ICASSP39728.2021.9414082.

[61] S. Petridis, T. Stafylakis, P. Ma, G. Tzimiropoulos, and M. Pantic, "Audio-Visual Speech Recognition with a Hybrid CTC/Attention Architecture," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, Dec. 2018, pp. 513–520. doi: 10.1109/SLT.2018.8639643.

[62] C.-T. Do, S. Zhang, and T. Hain, "Selective Adaptation of End-to-End Speech Recognition using Hybrid CTC/Attention Architecture for Noise Robustness," in 2020 28th European Signal Processing Conference (EUSIPCO), Jan. 2021, pp. 321–325. doi: 10.23919/Eusipco47968.2020.9287836.

[63] H. Inaguma and T. Kawahara, "VAD-free Streaming Hybrid CTC/Attention ASR for Unsegmented Recording," *arXiv:2107.07509 [cs, eess]*, Jul. 2021, Accessed: Dec. 25, 2021. [Online]. Available: http://arxiv.org/abs/2107.07509

[64] K. Deng, S. Cao, Y. Zhang, and L. Ma, "Improving Hybrid CTC/Attention End-to-end Speech Recognition with Pretrained Acoustic and Language Model," arXiv:2112.07254 [cs, eess], Dec. 2021, Accessed: Dec. 25, 2021. [Online]. Available: http://arxiv.org/abs/2112.07254

[65] L. Kürzinger, E. R. C. Rosas, L. Li, T. Watzel, and G. Rigoll, "Audio Adversarial Examples for Robust Hybrid CTC/Attention Speech Recognition," *arXiv:2007.10723 [cs, eess]*, Jul. 2020, Accessed: Dec. 25, 2021. [Online]. Available: http://arxiv.org/abs/2007.10723

[66] L. Wu, T. Li, L. Wang, and Y. Yan, "Improving Hybrid CTC/Attention Architecture with Time-Restricted Self-Attention CTC for End-to-End Speech Recognition," Applied Sciences, vol. 9, no. 21, p. 4639, Oct. 2019, doi: 10.3390/app9214639.

[67] M. Nie and Z. Lei, "Hybrid CTC/Attention Architecture with Self-Attention and Convolution Hybrid Encoder for Speech Recognition," J. Phys.: Conf. Ser., vol. 1549, no. 5, p. 052034, Jun. 2020, doi: 10.1088/1742-6596/1549/5/052034.

[68] Z. Yuan, Z. Lyu, J. Li, and X. Zhou, “An improved hybrid CTC-Attention model for speech recognition,” arXiv:1810.12020 [cs, eess], Nov. 2018, Accessed: Dec. 25, 2021. [Online]. Available: http://arxiv.org/abs/1810.12020

[69] H. Miao, G. Cheng, P. Zhang, and Y. Yan, “Online Hybrid CTC/Attention End-to-End Automatic Speech Recognition Architecture,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1452–1465, 2020, doi: 10.1109/TASLP.2020.2987752.

[70] F. Rudzicz, “Adjusting dysarthric speech signals to be more intelligible,” *Computer Speech & Language*, vol. 27, no. 6, pp. 1163–1177, Sep. 2013, doi: 10.1016/j.csl.2012.11.001.

[71] K. T. Mengistu and F. Rudzicz, “Adapting acoustic and lexical models to dysarthric speech,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 4924–4927. doi: 10.1109/ICASSP.2011.5947460.

[72] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, “Speech Recognition Using Deep Neural Networks: A Systematic Review,” *IEEE Access*, vol. 7, pp. 19143–19165, 2019, doi: 10.1109/ACCESS.2019.2896880.

[73] K. Hux, J. Rankin-Erickson, N. Manasse, and E. Lauritzen, “Accuracy of three speech recognition systems: Case study of dysarthric speech,” *Augmentative and Alternative Communication: AAC*, vol. 16, no. 3, p. 186, Sep. 2000.

[74] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. Huang, K. Watkin, and S. Frame. Dysarthric speech database for universal access research. In Proceedings of Interspeech, pages 22–26, Brisbane, Australia, 2008.

[75] D. S. Park *et al.*, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” *Interspeech 2019*, pp. 2613–2617, Sep. 2019, doi: 10.21437/Interspeech.2019-2680.

[76] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 4960–4964. doi: 10.1109/ICASSP.2016.7472621.

[77] A. Bittar and P. N. Garner, “A Bayesian Interpretation of the Light Gated Recurrent Unit,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 2965–2969. doi: 10.1109/ICASSP39728.2021.9414259.

[78] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” *arXiv:1409.0473 [cs, stat]*, May 2016, Accessed: Jan. 03, 2022. [Online]. Available: http://arxiv.org/abs/1409.0473

[79] A. Galassi, M. Lippi, and P. Torroni, “Attention in Natural Language Processing,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021, doi: 10.1109/TNNLS.2020.3019893.

[80] K. Xu et al., “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,” arXiv:1502.03044 [cs], Apr. 2016, Accessed: Jan. 03, 2022. [Online]. Available: http://arxiv.org/abs/1502.03044

[81] S. K. Mahata, D. Das, and S. Bandyopadhyay, “MTIL2017: Machine Translation Using Recurrent Neural Network on Statistical Machine Translation,” *Journal of Intelligent Systems*, vol. 28, no. 3, pp. 447–453, 2019, doi: 10.1515/jisys-2018-0016.

[82] B. Zhou, “Statistical Machine Translation for Speech: A Perspective on Structures, Learning, and Decoding,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1180–1202, May 2013, doi: 10.1109/JPROC.2013.2249491.

[83] N. Kalchbrenner and P. Blunsom, “Recurrent Continuous Translation Models,” in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, Oct. 2013, pp. 1700–1709. Accessed: Jan. 03, 2022. [Online]. Available: https://aclanthology.org/D13-1176

[84] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” *arXiv:1409.3215 [cs]*, Dec. 2014, Accessed: Jan. 03, 2022. [Online]. Available: http://arxiv.org/abs/1409.3215

[85] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches,” *arXiv:1409.1259 [cs, stat]*, Oct. 2014, Accessed: Jan. 03, 2022. [Online]. Available: http://arxiv.org/abs/1409.1259

[86] A. Graves and N. Jaitly, “Towards End-To-End Speech Recognition with Recurrent Neural Networks,” in *Proceedings of the 31st International Conference on Machine Learning*, Bejing, China, Jun. 2014, vol. 32, no. 2, pp. 1764–1772. [Online]. Available: https://proceedings.mlr.press/v32/graves14.html

[87] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving Language Understanding by Generative Pre-Training,” p. 12.

[88] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv:1810.04805 [cs]*, May 2019, Accessed: Jan. 03, 2022. [Online]. Available: http://arxiv.org/abs/1810.04805

[89] R. X. Gao and R. Yan, “Non-stationary signal processing for bearing health monitoring,” *IJMR*, vol. 1, no. 1, p. 18, 2006, doi: 10.1504/IJMR.2006.010701.

[90] .-C. Liu, F.-Y. Leu, G.-L. Lin, and H. Susanto, “An MFCC-based text-independent speaker identification system for access control,” Concurrency and Computation: Practice and Experience, vol. 30, no. 2, p. e4255, 2018, doi: 10.1002/cpe.4255.

[91] G. Chen, C. Parada, and G. Heigold, “Small-footprint keyword spotting using deep neural networks,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 4087–4091. doi: 10.1109/ICASSP.2014.6854370.

[92] “Snowboy, a Customizable Hotword Detection Engine — Snowboy 1.0.0 documentation.” http://docs.kitt.ai/snowboy/ (accessed Mar. 01, 2021).

[93] V. Z. Këpuska and T. B. Klein, “A novel Wake-Up-Word speech recognition system, Wake-Up-Word recognition task, technology and evaluation,” *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 12, pp. e2772–e2789, Dec. 2009, doi: 10.1016/j.na.2009.06.089.

[94] Z. Wang, X. Li, and J. Zhou, “Small-footprint Keyword Spotting Using Deep Neural Network and Connectionist Temporal Classifier,” *arXiv:1709.03665 [cs]*, Sep. 2017, Accessed: Jan. 03, 2022. [Online]. Available: http://arxiv.org/abs/1709.03665

[95] X. Shu, X. Xiao, Y. Long, G. Zeng, and Y. Zhong, “Background modeling methods based on two-stage strategy,” in *IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012)*, Dec. 2012, pp. 1–4. doi: 10.1049/cp.2012.2328.

[96] M. Wu *et al.*, “Monophone-Based Background Modeling for Two-Stage On-Device Wake Word Detection,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5494–5498. doi: 10.1109/ICASSP.2018.8462227.

[97] S. Mohamad Suhaili, N. Salim, and M. N. Jambli, “Service chatbots: A systematic review,” *Expert Systems with Applications*, vol. 184, p. 115461, Dec. 2021, doi: 10.1016/j.eswa.2021.115461.

[98] S. Tabibian, “A survey on structured discriminative spoken keyword spotting,” Artif Intell Rev, vol. 53, no. 4, pp. 2483–2520, Apr. 2020, doi: 10.1007/s10462-019-09739-y.

[99] M. J. F. Gales, K. M. Knill, A. Ragni, and S. P. Rath, “Speech Recognition and Keyword Spotting for Low Resource Languages: Babel Project Research at Cued.”

[100] G. Deekshitha and L. Mary, “Multilingual spoken term detection: a review,” *Int J Speech Technol*, vol. 23, no. 3, pp. 653–667, Sep. 2020, doi: 10.1007/s10772-020-09732-9.

[101] L. Pandey and R. M. Hegde, “Keyword Spotting in Continuous Speech Using Spectral and Prosodic Information Fusion,” *Circuits Syst Signal Process*, vol. 38, no. 6, pp. 2767–2791, Jun. 2019, doi: 10.1007/s00034-018-0990-6.

[102] S. Li, G. Li, J. Han, and T. Zhi, “Overview of speech keyword recognition technology,” *J. Phys.: Conf. Ser.*, vol. 1827, no. 1, p. 012013, Mar. 2021, doi: 10.1088/1742-6596/1827/1/012013.

[103] J. L. K. E. Fendji, D. M. Tala, B. O. Yenke, and M. Atemkeng, “Automatic Speech Recognition using limited vocabulary: A survey,” arXiv:2108.10254 [cs, eess], Aug. 2021, Accessed: Jan. 04, 2022. [Online]. Available: http://arxiv.org/abs/2108.10254

[104] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989, doi: 10.1109/5.18626.

[105] P. D. Polur and G. E. Miller, “Experiments with fast Fourier transform, linear predictive and cepstral coefficients in dysarthric speech recognition algorithms using hidden Markov model,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 4, pp. 558–561, 2005, doi: 10.1109/TNSRE.2005.856074.

[106] Z. Özcan and T. Kayıkçıoğlu, “Evaluating MFCC-based speaker identification systems with data envelopment analysis,” *Expert Systems with Applications*, vol. 168, p. 114448, Apr. 2021, doi: 10.1016/j.eswa.2020.114448.

[107] C. Jose, Y. Mishchenko, T. Senechal, A. Shah, A. Escott, and S. Vitaladevuni, “Accurate Detection of Wake Word Start and End Using a CNN,” arXiv:2008.03790 [cs, eess], Aug. 2020, Accessed: Nov. 15, 2021. [Online]. Available: http://arxiv.org/abs/2008.03790