Keyword spotting in multi-player voice driven games for children

Harshavardhan Sundar[2], *Jill Fain Lehman*[1], *Rita Singh*[1]

[1]Disney Research Labs, Pittsburgh, PA, USA.
[2]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

`suharsha@cs.cmu.edu, jill.lehman@disneyresearch.com, rsingh@cs.cmu.edu`

Abstract

Word spotting, or keyword identification, is a highly challenging task when there are multiple speakers speaking simultaneously. In the case of a game being controlled by children solely through voice, the task becomes extremely difficult. Children, unlike adults, typically do not await their turn to speak in an orderly fashion. They interrupt and shout at arbitrary times, speak or say things that are not within the purview of the game vocabulary, arbitrarily stretch, contract, distort or rapid-repeat words, *and* do not stay in one location either horizontally or vertically. Consequently, standard state-of-art keyword spotting systems that work admirably for adults in multiple keyword settings, fail to perform well even in a basic two-word vocabulary keyword spotting task in the case of children. This paper highlights the issues with keyword spotting using a simple two-word game played by children of different age groups, and gives quantitative performance assessments using a novel keyword spotting technique that is especially suited to such scenarios.

Index Terms: Word spotting, Keyword identification, Children's speech, Children's games, Distant speech recognition, Voice driven games.

1. Introduction

Keyword spotting in continuous speech involves detecting whether and where in a recording a specified word occurs. Currently, depending on the task setting, there are four categories of approaches that are mainstream for keyword spotting. The most basic approach is to simply set the key-terms in opposition to a generic garbage model and apply a likelihood ratio test to identify the keywords [1, 2]. The second comprises performing phoneme (or syllable, or other sub-word unit) recognition. Keyword spotting is then done by searching for specific sequences of phonemes in a recording and coalescing them into words [3]. The third category comprises performing large-vocabulary recognition with a language model, and searching for the desired key-terms in the ASR system lattices [4].

In the fourth set of techniques, spoken examples of the keywords are used to build specific word detectors. We refer to these as *example-based methods*. Example-based methods model each keyword to be spotted in its *entirety*. While phoneme-based methods are flexible, example-based methods are generally more accurate or faster, *e.g.* [5, 6, 7]. Many example-based methods basically mimic the phoneme-based methods in that they attempt to first derive a phoneme sequence or lattice, which is re-evaluated with the phonetic models in an ASR system to generate word identities [8]. Such example-based techniques also include those based on neural networks. Lately, *bidirectional* neural networks [9], especially bidirectional long-short-term memory (BLSTM) neural networks have been employed with particular success for word spotting tasks [10]. Word spotting with BLSTMs has focused on text- or phoneme-sequence-based word specification: the word spotter either scans the phoneme lattice generated with the BLSTMs for the specified words [11, 12], or uses a second-level discriminative classifier that employs features derived from the lattice to detect the words [13].

In this paper we address the problem of keyword spotting in the setting of a multiplayer children's game. This setting is challenging for multiple reasons. First, there is a much larger inherent variability in the quality of children's voices than for adults. This is related to physiological issues [14]. In children, the length of the vocal cord, and the shape of the vocal tract changes more rapidly with age than in adults. As a result, marked changes in voice characteristics are seen with smaller age differentials in children [15], and a system cannot be trivially set up to work optimally for all age groups with one setting. In addition, through childhood, the length of the vocal cords is not at the optimal adult length needed to discriminatively pronounce each sound in the language. Children therefore enunciate sounds in a much more varied fashion than adults do [16]. In fact, children's speech becomes clearer with increasing age, although the pitch is higher on average in child speech, and there is more energy in higher frequencies as compared to adult speech. For all these reasons, training keyword detectors for children's voices is expected to be an inherently difficult task.

The second reason is related to stylistic issues that correlate with developmental and emotional factors [17, 18, 19]. Children also generally do not adhere to turn-taking rules, and even within a restricted vocabulary game, they often speak concurrently. In the keyword-spotting scenario, multiple keywords could be spoken by multiple children at the same time. Thus any effective keyword-spotting algorithm used for a multiplayer scenario must also necessarily address the case where keywords are simultaneously uttered by multiple players. The stylistic issues are exemplified clearly in the context of a game called "Mole Madness" in this paper. Mole Madness is a two-dimensional side-scroller game, similar to video games like Super Mario Bros®. The goal of the game is to move a mole character through its environment, capturing food and avoiding dangers (see Figure 1). Two children work cooperatively to move the mole, one child effecting horizontal movement with the word GO, and the other controlling vertical movement with JUMP. Without speech, the mole simply falls to the ground and spins in place.

On the surface, the game presents a simple two-keyword spotting task that should be easy to address with any of the existing techniques mentioned above that are known to work well for adult voices. However, stylistic issues with children's voices, especially when trying to stress function word sounds [20], seriously confound the solutions. In addition to the ex-

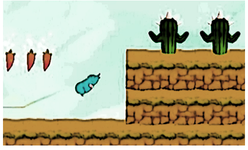
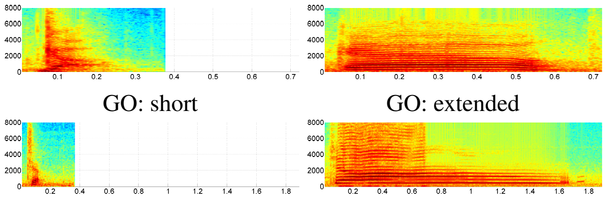


Figure 1: A screen shot of Mole Madness. The mole (blue) must GO and JUMP to scale the wall and avoid the cactii.

pected physiologically-variable pronunciations of participants (ages 4 to 10), the following phenomena occur throughout:

1. Extreme variability in duration: On one hand, we find the word "go" so contracted that multiple repetitions are included in one rapid-fire instance, *e.g.* GOGGOGGO. Note the skipped syllable. On the other hand, we also find the word extremely prolonged in duration, e.g GOOOOOOOOOOOOOOO. Figure 2 presents some examples.
2. Keyword merging: Under time pressure and in an excited state, children will subsume each other's roles, producing garbled, merged forms of the keywords *e.g.* GJMP, JUMPGOOJJJJJUMP, etc.
3. Speech-on-speech: Areas of the game environment are designed to make the children work together to acquire a reward or overcome an obstacle, creating instances of moderate to near-perfect overlap in speech. In such cases the system must be able to detect that both words were spoken. This is the main motivation for presenting the algorithm in this paper.
4. Voice tremors and acoustic distortion: Jumping vertically and horizontally while screaming causes acoustic distortions in voice, more in adults but even in children.
5. Out-of-vocabulary words: Despite the desire to move the mole quickly and achieve a high score, children will, nevertheless, participate in social and strategic side-talk [21], *e.g.* SAY GO, I SAID JUMP, NICE MOVE, etc. This introduces out-of-vocabulary words that are not merely resultant from mergings or contractions.

In the following sections we present an algorithm for keyword spotting that is best suited for multi-person game scenarios when overlaps are expected between spoken instances of keywords. We compare the algorithm to state-of-art BLSTMs modified for keyword spotting in a multi-person scenario, and evaluate its performance in different environmental noise conditions. Experiments show that the algorithm outperforms conventional techniques in scenarios where the keywords are phonemically and durationally altered or distorted, and especially when the keywords are overlapped.



GO: short

GO: extended

JUMP: short

JUMP: extended

Figure 2: Spectrograms showing durational variability of keywords in Mole Madness.

2. Maximum-likelihood wordspotting with heavy-tailed distributions

Let us consider a task where N keywords $\{w_1, w_2, \ldots, w_N\}$ are to be spotted. The recording environment includes multiple speakers, any of who may speak the keywords. Speakers may also speak concurrently. The task is to determine which of the keywords were spoken. If multiple keywords are concurrently spoken (possibly with partial overlap), all of them must be identified.

We process the incoming speech in overlapping blocks. The width of the blocks and the overlap between adjacent blocks is optimized empirically. The audio in any block is parameterized into a sequence of feature vectors (Mel-frequency cepstral vectors in our case), each representing one *frame* of the block. We represent the sequence of feature vectors derived from any audio block as F and the individual feature vectors in it by f.

Recalling that keywords may be spoken concurrently, one or more of the keywords may occcur within each block. We view every combination of keywords as a separate class of events. Since there are N keywords, there are 2^N classes, representing the 2^N possible combinations of keywords. For notational convenience, we can represent the combination of keywords in any block through an N-bit indicator $Z = z_1 z_2 \cdots z_N$, where each z_i is a single bit indicating the occurrence or absence of the i^{th} keyword. Z can take 2^N possible values, each indicating one of the 2^N possible combinations of keywords, and indexes the classes.

The features F derived from the audio for each combination of keywords Z can be expected to have their own distinctive statistical signature. In practice, however, there is potentially infinite variation in the asynchrony between the utterance of the individual words in the combination, resulting in a large variation in the temporal patterns in F. To effectively capture the asynchrony, we assume the generative model shown in Figure 3 for F. The model assumes every vector f in F to be statistically independent of every other vector. In order to generate any vector f, first the class Z is drawn from a prior distribution $P(Z)$; subsequently the vector f is drawn from the class-conditional distribution $P(f|Z)$. In effect, the model assumes that every vector in F may have been drawn from a different class. The model implicitly accounts for the fact that when multiple keywords are spoken concurrently, different combinations of these keywords may be active in any individual frame due to the asynchrony between the keywords. This results in the following probability distribution for F

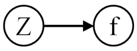


Figure 3: Generative model for F

$$P(F) = \prod_{f \in F} \sum_{Z} P(Z) P(f|Z)$$

Note that in this model $P(Z)$ effectively represents the expected fraction of feature vectors in F that were obtained from the combination Z. If the class-specific distributions $P(f|Z)$ are known, the task of identifying the set of keywords active in any block thus reduces to identifying the set of Zs that are most dominant in the block, *i.e.*, the Zs corresponding to the largest $P(Z)$ values.

This leads to the following algorithm for identifying keywords.

- In a training phase, we learn a model $P(f|Z)$ for the distribution of feature vectors recorded under every combination Z of keywords from appropriate collections of training data.
- In the test phase, for every block B of test-data audio represented by the features F_B, we estimate the probability distribution $P(F_B) = \sum_Z P_B(Z)P(f|Z)$, where we have used the subscript B in $P_B(Z)$ to denote that the distribution is specific to block B. Since $P(f|Z)$ is already known (from the training phase), only $P_B(Z)$ must be estimated. We use a simple EM algorithm to obtain a maximum-likelihood estimate of $P_B(Z)$. Subsequently, we compute the probability that the i^{th} keyword was spoken as the sum of the $P_B(Z)$ values for all Zs in which the keyword was included.

$$P_B(i) = \sum_{Z:z_i=1} P_B(Z) \qquad (1)$$

$P_B(i)$ estimates the fraction of frames in F_B in which the i^{th} keyword was active. The keywords with the largest $P_B(i)$ values are returned as candidates for the block.

The procedure above places no explicit constraint on the actual combination-specific distributions $P(f|Z)$. Although, in principle, conventional distributions such as Gaussian mixture models may be employed, in mixed-speech scenarios, we have found that the features are most effectively modeled by heavy-tailed distributions [22]. In this work we have found it most effective to model $P(f|Z)$ as a mixture of Student's-*t* distributions. The parameters of these distributions can be learned from training examples, as given in [22]. Distributions for Z values representing multiple concurrent keywords may be learned from synthetic mixtures of the keywords. We refer to the mixture-Student's-*t* based models as M-TMMs. As a comparator we have also modeled $P(f|Z)$ by GMMs. We refer to this model as M-GMM.

For large values of N, the total number 2^N of keyword combinations can become very large. In these situations, we have found it effective to assume that no more than one (or a small number K) of the keywords is spoken at any time, *i.e.*, assume that individual frames are dominated either by a single keyword, or no keyword at all. For the results reported in this paper, however, $N = 2$, and the entire set of 2^N possible combinations are considered.

3. The BLSTM comparator

As a comparator to our proposed method, we evaluated bidirectional long-short-term memory networks (BLSTMs) for the word-spotting task. These have been known to work well for children's speech [23]. BLSTMs capture both *causal* and *anti-causal* recurrence, through a network that has both a *forward* and a *backward* component; classification is performed using information from both the networks. A BLSTM network is a bidirectional recurrent neural network with LSTM cells in its hidden layers. An LSTM neuron or *cell* is a unit comprising a central memory unit and *gates* that determine when it must consider an input, when it must remember a value, and when it should output its value [24]. BLSTMs have been shown to be highly effective at a variety of speech recognition tasks, including word spotting, and are considered state-of-art in many of these tasks [25].

In our setup, we employed a BLSTM network to directly classify each block of incoming speech into one of 2^N classes, corresponding to the 2^N possible combinations of keywords. The network is trained with several instances from each of these 2^N conditions. The output layer has 2^N soft-max neurons, the outputs of which may be interpreted as $P(Z|F)$, where Z is defined as in Section 2. In the test phase, each block of incoming data is processed by the BLSTM network. The outputs are combined according to Equation 1 to compute probabilities for the individual words.

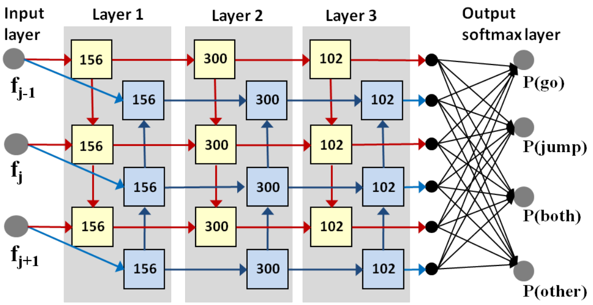


Figure 4: A schematic of the BLSTM network architecture. Forward blocks of LSTM neurons are colored yellow and backward ones are colored blue. The number of memory neurons used is indicated within each block.

The configuration of the network used in our experiments was identical to the one used for word recognition in the 2013 2^{nd} ChiME Challenge track-1 [26]: the network had three hidden layers (indicated as Layers 1, 2 and 3 in Fig. 4) containing 156, 300 and 102 neurons, respectively. The full architecture is shown in Fig. 4. We used CURRENNT [27], a publicly available CUDA-based implementation. A learning rate of 10^{-5} and a momentum of 0.9 is used in a stochastic gradient descent method for optimizing the network parameters during training.

4. Experimental results

We evaluated three algorithms: both the M-TMM and M-GMM variants of the proposed algorithm and BLSTMs, on Mole Madness data collected in-house at Disney Research. The data comprise recordings from 34 distinct pairs of children between the ages of 4 and 10 years (mean age 7 years, standard deviation = 2.01 years, male-female ratio of 3:2). Each pair of children played twice, once as GO and once as JUMP. For data collection, the application did not use a speech recognizer to respond to the children's voices. Instead, the players controlled the mole through "Wiimotes" [28], while simultaneously uttering the required keywords. Each child wore a close-talking microphone on his or her shoulder, with data recorded at a sampling rate of 16 kHz.

The Mole Madness database contains 1723 isolated utterances of GO and 1214 utterances of JUMP. Tracks from each child were independently annotated. Overlaps can thus be determined by matching time-stamps between the tracks. Data segments which did not contain these keywords are annotated as "Background". For our experiments, we used 1200 non-overlapping GO, 800 instances of non-overlapping JUMP and 254 segments of Background.

The test set for evaluating performance on *non-overlapped* speech comprised 523 instances of GO, 414 instances of JUMP and 230 instances of Background. The sets of children from which the training and testing data are chosen were mutually exclusive. The test set for evaluating performance on *overlapped* speech comprised 523 instances of concurrent instances of GO and JUMP being spoken together. These latter count as positives for both GO and JUMP. To test the performance under

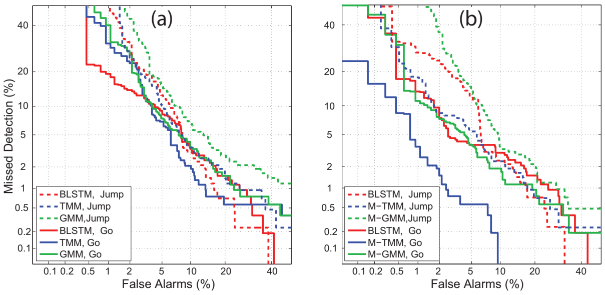


Figure 5: (a): Performance obtained within a hypothesis-testing framework. (b) DET Curves for M-TMM, M-GMM and BLSTMs under the proposed framework.

different SNR conditions, we synthetically added babble noise (recorded in a gaming parlor) to the test set.

In all experiments, the features used were 39-dimensional Mel-Frequency Cepstra (including delta and acceleration coefficients). Performance was evaluated by generating Detection Error Threshold (DET) curves for each. The DET curve plots missed-detection (MD) vs. false-alarm (FA). We also computed the *detection cost function* (DCF) as a scalar quantitative metric of performance. The DCF is a weighted combination of false-alarm and missed-detection. Because the cost of FA is higher than MD in Mole Madness, we set the DCF to be the weighted combination: $0.3 * P(MD) + 0.7 * P(FA)$, where $P()$ denotes probability.

We note that keyword spotting may also be viewed as a conventional hypothesis-testing problem. To establish the baseline for this task, we first performed spotting within a hypothesis-testing framework. Two models were trained for each word: one for the word itself, and the other for the negative class using all data that did not include the word. For the spotting task, a log likelihood score was obtained using both models. Both TMMs and GMMs were used for this purpose. TMMs and GMMs used mixtures of 64 component distributions. Since TMMs and GMMs do not incorporate temporal structure, we also trained a third spotter using BLSTMs. Here, for each word we trained a BLSTM to distinguish between the word and the absence of the word. Figure 5a shows the DET curves for all three spotters, for both words in our test set. We note that there is no significant difference in performance between the three. In particular, the temporal structure captured by the BLSTM brings no significant benefit, showing clearly that even the BLSTM is unable to capture the large variations in temporal structure in children's speech under game conditions.

Figure 5b shows the performance obtained using our proposed method on clean test instances of both words. We show the performance with M-TMMs, M-GMMs, and BLSTMS implemented as described in Section 3. We note directly that explicitly modeling the condition where words overlap improves the performance of BLSTMs. However the best performance is obtained using M-TMMs. The combination of the proposed maximum-likelihood framework and the use of mixtures of Student's-*t* distributions to model the words results in the best performance.

Figure 6 shows the performance obtained using the proposed method on test instances that were corrupted by background (babble) noise to various signal levels. The upper and lower panels show, separately, the performance on non-overlapping and over-lapping instances of the words.

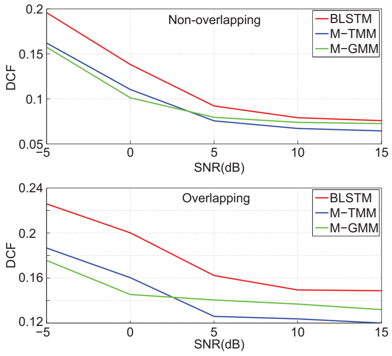


Figure 6: Top: Average DCF as a function of SNR, on non-overlapping instances of words corrupted by background gaming-parlor noise, using the M-TMMs, M-GMMs, and BLSTMs. Bottom: DCF as a function of SNR on overlapping instances of the words, for the same three methods.

5. Discussion of Results

We note firstly that all results are poorer than we would expect for such small vocabularies. The performances reported here are, in fact, better than those obtained with standard commercial speech recognizers (although not reported here) and other ASR based recognizers [29]. Children's speech, particularly from *excited* children, is just fundamentally extremely difficult to recognize for all the reasons mentioned above.

The results show interesting patterns. The two-class hypothesis-testing framework of Figure 5a performs consistently worse than all variants of the proposed method. Of particular note is that the TMM and GMM frameworks, which model feature vectors as IID, do not perform significantly worse than BLSTMs – the temporal structure captured by the BLSTM brings no significant benefit. The large variations in temporal structure in children's speech under game conditions are difficult to model even with models such as BLSTMs with many parameters. This is in opposition to all other experiments on adult speech, where the BLSTM is shown to distinctly result in large improvements over more detailed models, including HMMs.

The proposed maximum-likelihood mechanism which estimates the contributions of the individual classes to the test data by "fitting" the mixture of class distributions to the data seems to significantly outperform the more conventional hypothesis-testing framework on this data, suggesting that such approaches (which discard temporal structure) can in fact be used effectively in some situations where the temporal structure in the data may be obscured by intrinsic or extrinsic effects.

A secondary observation is that the mixtures of Student's-*t* distributions result in better performance than mixtures of Gaussians. This indicates that children's speech may include a larger fraction of *episodic* phenomena, which are better modeled by heavy-tailed distributions [30].

As can be seen in Figure 6, the performance of all classifiers degrades with noise; however the performance degradation due to low SNRs is more predominant in BLSTMs than in the proposed approach, particularly when Student's-*t* distributions are used. The ability of the heavy-tailed Student's-*t* to effectively represent outliers, which are more common in noisy speech, apparently provides significant benefit under these conditions. We believe that at least a part of the gains are due to the limited amount of training data, and that with larger amounts of data the differences may diminish. Investigation of the effect of increasing the amount and variety of data, game vocabulary, levels of excitement, etc. remain areas of current and future work.

6. References

[1] Mitchel Weintraub. LVCSR log-likelihood ratio scoring for keyword spotting. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP-95)*, volume 1, pages 297–300. IEEE, 1995.

[2] Igor Szöke, Petr Schwarz, Pavel Matejka, Lukas Burget, Martin Karafiát, Michal Fapso, and Jan Cernockỳ. Comparison of keyword spotting approaches for informal continuous speech. In *Interspeech*, pages 633–636, 2005.

[3] Richard C Rose and Douglas B Paul. A hidden markov model based keyword recognition system. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP-90)*, pages 129–132. IEEE, 1990.

[4] Petr Motlicek, Fabio Valente, and Igor Szoke. Improving acoustic based keyword spotting using LVCSR lattices. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP2012)*, pages 4413–4416. IEEE, 2012.

[5] T. Ezzat and T. Poggio. Discriminative word-spotting using ordered spectro-temporal patch features. In *SAPA workshop (INTERSPEECH)*, Brisbane, Australia, 2008.

[6] S.Fernandez, A. Graves, and J. Schmidhuber. An application of recurrent neural networks to discriminative keyword spotting. In *Proc. ICANN*, page 220229, Porto, Portugal, 2007.

[7] A. Jansen and P. Niyogi. Point process models for spotting keywords in continuous speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(8):1457–1470, 2009.

[8] T. J. Hazen, W. Shen, and C. White. Query-by-example spoken term detection using phonetic posteriogram templates. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Merano, Italy, December 2009.

[9] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[10] F. A. Gers. Long short-term memory in recurrent neural networks. *PhD thesis*, pages 2673–2681, 2001.

[11] Martin Wöllmer, Florian Eyben, Alex Graves, Björn Schuller, and Gerhard Rigoll. Improving keyword spotting with a tandem BLSTM-DBN architecture. In *Advances in Nonlinear Speech Processing*, pages 68–75. Springer, Heidelberg, 2010.

[12] Martin Wöllmer, Florian Eyben, Björn Schuller, Y. Sun, T. Moosmayr, and N. Nguyen-Thien. Robust incar spelling recognition - a tandem BLSTM-HMM approach. In *Proc. of Interspeech, ISCA*, pages 2507–2510, Brighton, UK, 2009.

[13] Y. Sun, T. Bosch, and L. Boves. Hybrid HMM/BLSTM-RNN for robust speech recognition. In *Proceedings of the 13th International Donference on Text, Speech and Dialogue*, pages 400–407, Springer-Verlag, September 2010.

[14] Annerose Keilmann and Carl-Albert Bader. Development of aerodynamic aspects in children's voice. *International Journal of Pediatric Otorhinolaryngology*, 31(2):183–190, 1995.

[15] Leslie E Glaze, Diane M Bless, Paul Milenkovic, and Robin D Susser. Acoustic characteristics of children's voice. *Journal of Voice*, 2(4):312–319, 1988.

[16] Sungbok Lee, Alexandros Potamianos, and Shrikanth Narayanan. Acoustics of children's speech: Developmental changes of temporal and spectral parameters. *The Journal of the Acoustical Society of America*, 105(3):1455–1468, 1999.

[17] Bruce L Smith. Relationships between duration and temporal variability in childrens speech. *The Journal of the Acoustical Society of America*, 91(4):2165–2174, 1992.

[18] Harry Levin, Irene Silverman, and Boyce L Ford. Hesitations in children's speech during explanation and description. *Journal of Verbal Learning and Verbal Behavior*, 6(4):560–564, 1967.

[19] Stefan Steidl. *Automatic classification of emotion related user states in spontaneous children's speech*. Phd thesis, University of Erlangen-Nuremberg, Germany, 2009.

[20] Margaret Kehoe, Carol Stoel-Gammon, and Eugene H Buder. Acoustic correlates of stress in young children's speech. *Journal of Speech, Language, and Hearing Research*, 38(2):338–350, 1995.

[21] J. F. Lehman and S. Al Moubayed. Mole madness: A multi-child, fast-paced, speech-controlled game. In *AAAI Spring Symposium on Turn-taking and Coordination in Human-Machine Interaction*, 2015.

[22] Harshavardhan Sundar, Thippur V Sreenivas, and Walter Kellermann. Identification of active sources in single-channel convolutive mixtures using known source models. *IEEE Signal Processing Letters*, 20(2):153–156, 2013.

[23] Martin Wöllmer, Björn Schuller, Anton Batliner, Stefan Steidl, and Dino Seppi. Tandem decoding of children's speech for keyword detection in a child-robot interaction scenario. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(4):12, 2011.

[24] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[25] A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Phd thesis, Technische Universitt München, July 2008.

[26] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll. The Munich feature enhancement approach to the 2013 CHiME Challenge using BLSTM recurrent neural networks. In *Proc. 2nd CHiME Speech Separation and Recognition Challenge*, 2013.

[27] F. Weninger, J. Bergmann, and B. Schuller. Introducing CURRENNT - the Munich open-source CUDA recurrent neural network toolkit. *Journal of Machine Learning Research*, 2014.

[28] Wikipedia. `http://en.wikipedia.org/wiki/Wii_Remote`.

[29] P. Baljekar, J. F. Lehman, and R. Singh. Online word-spotting in continuous speech with recurrent neural networks. In *Spoken Language Technology Workshop*. IEEE, 2014.

[30] Deniz Gencaga. *Sequential Bayesian modeling of non-stationary non-Gaussian processes*. Phd thesis, Bogazici University, 2007.