Automatic dysfluency detection in dysarthric speech using deep belief networks

Stacey Oue[1], *Ricard Marxer*[2], *Frank Rudzicz*[1,3]

[1]Department of Computer Science, University of Toronto;
[2]Department of Computer Science, University of Sheffield;
[3]Toronto Rehabilitation Institute-UHN

stacey.oue@mail.utoronto.ca, r.marxer@sheffield.ac.uk, frank@cs.toronto.edu

Abstract

Dysarthria is a speech disorder caused by difficulties in controlling muscles, such as the tongue and lips, that are needed to produce speech. These differences in motor skills cause speech to be slurred, mumbled, and spoken relatively slowly, and can also increase the likelihood of dysfluency. This includes non-speech sounds, and 'stuttering', defined here as a disruption in the fluency of speech manifested by prolongations, stop-gaps, and repetitions. This paper investigates different types of input features used by deep neural networks (DNNs) to automatically detect repetition stuttering and non-speech dysfluencies within dysarthric speech. The experiments test the effects of dimensionality within Mel-frequency cepstral coefficients (MFCCs) and linear predictive cepstral coefficients (LPCCs), and explore the detection capabilities in dyarthric versus non-dysarthric speech. The results obtained using MFCC and LPCC features produced similar recognition accuracies; repetition stuttering in dysarthric speech was identified correctly at approximately 86% and 84% for non-dysarthric speech. Non-speech sounds were recognized with approximately 75% accuracy in dysarthric speakers.

Index Terms: Dysarthria, stuttering, non-speech dysfluency, DNN, MFCC, LPCC

1. Introduction

Many studies have researched ways to improve the intelligibility of dysarthric speech, including methods that targeted particular aspects of speech to modify. Kain *et al.* [1] implemented a system of transformations that focused strictly on mapping vowels from individuals with dysarthria to vowels more characteristic of non-dysarthric speech. Those experiments showed an intelligibility increase of 6%. In 2013, Rudzicz [2] proposed a method that added the correction of other pronunciation errors and adjusted tempo. Among a cohort of listeners unfamiliar with the speech of people with cerebral palsy, word recognition rates increased by 19.6%. Crucially, the Levenshtein-based detection of phoneme repetitions and non-speech dysfluencies in that work depended on full phoneme segmentation, which may itself be quite challenging for dysarthric speech.

Chee *et al.* [3] provided an overview of automatic stuttering detection, emphasizing its difficulty across a number of classification methods. Czyzewski *et al.* [4], e.g., implemented artificial neural networks (ANNs) and 'rough sets' to detect three types of 'stuttering': stop-gaps, vowel prolongations, and syllable repetitions, obtaining accuracies up to 73.25% with ANNs and 91% with rough sets. Wiśniewski *et al.* [5, 6] performed two studies that used hidden Markov models with Mel-frequency cepstral coefficients (MFCCs) to detect stuttering. The first focused on both prolongation of fricative phonemes and blockades with repetition of stop phonemes that produced an accuracy of 70% [5]; the second strictly focused on prolongation of fricative phonemes and found an improvement in accuracy to approximately 80% [6].

Rath investigated modifications to MFCC feature vectors in speaker adaptation using deep neural networks (DNNs) [7], obtaining 3% improvements over Gaussian mixture models (GMMs) baselines. Across various types of speech features, deep learning has shown considerable improvements across several areas of speech recognition [8], compared with traditional techniques such as hidden Markov models. Here, we compare MFCCs (which are the most commonly used feature set in this domain [3]) and linear predictive cepstral coefficients (LPCCs), which are another popular but less utilized feature set. An exception was Chee *et al.* [9], who applied LPCCs with k-nearest-neighbors and linear discriminant analysis classifiers to automatically detect prolongations and repetition stutters, with recognition accuracy up to 89.77%. In the related field of automatic speech recognition ()ASR), MFCCs have consistently generated better results than LPCCs [10, 11]; to see if this trend extends to the domain of dysfluency detection, we compare these feature types with DNNs.

2. Methodology

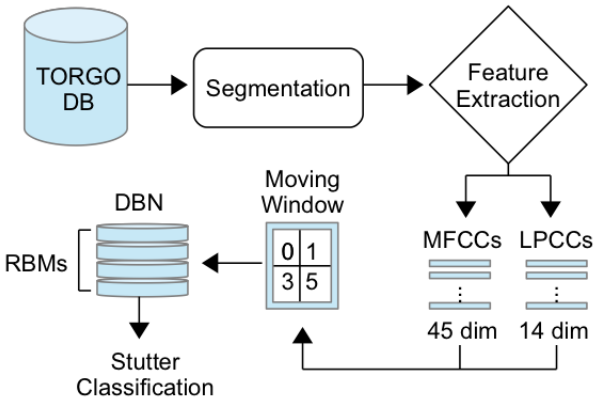

Figure 1: *Overview of automatic stuttering detection method.*

2.1. Data

The TORGO database [12] was created by a collaboration between the departments of Computer Science and Speech-

SLPAT 2015, 6th Workshop on Speech and Language Processing for Assistive Technologies, pages 60–64,
Dresden, Germany, 11 September, 2015.

Language Pathology at the University of Toronto, and the Holland-Bloorview Kids Rehab hospital. The corpus consists of recordings from seven participants, three females and four males ranging in age from 16 to 50, diagnosed with cerebral palsy or amyotrophic lateral sclerosis. Additionally, there are recordings from seven control speakers matched for age and gender. A combination of non-words, short words, restricted sentences, and unrestricted sentences were recorded by all participants with a 16 kHz sampling frequency using two microphones. The database also includes articulatory measurements using electromagnetic articulography, which is not used here.

2.2. Segmentation

Segmentation was performed manually by listening to the recorded speech samples in the TORGO database and marking the start and end times of each occurrence of stutters. Only a single type of 'stuttering' dysfluency is considered here, specifically repetition-type stutters (Table 1), since these are more difficult to detect than prolongations and stop-gaps [4].

Table 1: *Repetition Types*

Repetition Type	Example
Part of a word	wh-wh-what time is it?
Whole word	what-what-what time is it?
Phrase	what time what time is it?

For the analysis of non-speech dysfluencies we employed the phonetic transcriptions provided with the TORGO database. In such transcriptions, non-phonetic segments are marked with the label *noi* (noise).

2.3. Feature extraction

After segmentation, speech data were parameterized into an input form suitable for use by a DNN classifier (Figure 2), as described below.

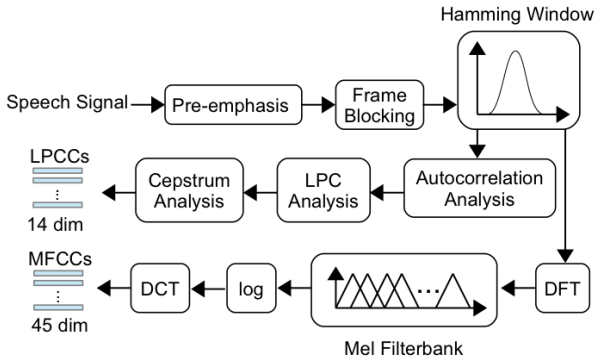

Figure 2: *MFCC and LPCC feature extraction overview.*

2.3.1. MFCC features

The MFCC input feature baseline consists of 13 cepstral coefficients in addition to the 0^{th} cepstral coefficient, energy, δ, and $\delta\delta$ coefficients. There is no pre-emphasis performed on these features. Since speech samples are constantly changing, we use frame blocking to analyze the signal in small time frames such that it becomes near stationary. The speech signals are cut into 25 ms frames with a frame step of 10 ms. We use a Hamming window to calculate the MFCC features, where the coefficients are found given Equation 1 (N is equal to window size minus one, in this case $N = 399$).

$$w(n) = 0.54 - 46cos(2\pi\frac{n}{N}), \qquad 0 \leq n \leq N \qquad (1)$$

To detect the different frequencies in the signal, the power spectrum is calculated using the discrete Fourier transform (DFT). The Mel filterbank then sums the energy in each filter, obtaining 29 uniformly-distributed triangular filters. The discrete cosine transform (DCT) is then applied to the log-filterbank energies to obtain the MFCCs. The purpose of the DCT is to decorrelate the overlapping filterbanks.

2.4. LPCC features

The LPCC features include 13 coefficients followed by the energy coefficient. LPCCs are more vulnerable to noise than MFCCs, so the speech signal is flattened before processing to avoid additive noise error. This is accomplished by pre-emphasis, a first order high-pass filter is applied to the speech signal as in

$$H(z) = 1 - az^{-1}, \qquad a = e^{-\frac{100\pi}{16000}} = 0.9806. \qquad (2)$$

Frame blocking and the Hamming window are applied to the LPCC feature space with the same parameters as for MFCCs (i.e., frame blocking 25 ms, 50% frame overlap and frame step 10 ms). This is followed by LPC analysis that estimates the coefficients by using the autocorrelation method to obtain fundamental frequency, pitch, and repeating patterns in the speech signal, before cepstral analysis is performed.

2.5. Feature modulation

We explored increasing the dimension of the input features used by the DNN due to the fact that DNNs are robust to larger input dimensions. The frequently-used hidden Markov model with Gaussian mixture output densities can become subject to error in parameter estimation, even with a slight increase in the input dimensions. The concept of a moving window is implemented to create inputs with larger dimensions. The moving window considers frames before and after the current frame. For example, a window of size $\pm x$ takes the x consecutive frames preceding and following the current frame and combines them into a single input vector (Figure 3 provides a visual representation of a moving window of size ± 1).

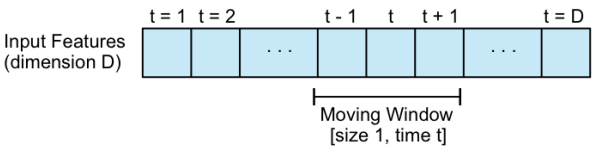

Figure 3: *Moving window of size ± 1.*

The dimensions of the input features are provided in Table 2. The baseline number of MFCCs and LPCCs are 45 and 14, respectively. The purpose of the moving window is to exploit the DNN's ability to use higher-dimensional input feature vectors to achieve better classification results by integrating contextual information.

Table 2: *Input feature dimensions*

Moving window size	0	±1	±3	±5
MFCC input dimension	45	135	315	495
LPCC input dimension	14	42	98	154

2.6. Classification

We use the deep neural network implementation of Tanaka and Okutomi [13] for stuttering classification. Four pre-trained Bernoulli-Bernoulli restricted Boltzman machines (RBMs) plus a decision layer are stacked to form a deep belief network (DBN), to create a DBN-DNN classifier (Figure 4). The RBMs are pre-trained in an unsupervised way using contrastive divergence. Once the DBN is initialized with the pre-trained RBMs, we fine-tune the DBN with a supervised learning method based on reducing error in the classification of, alternatively, stuttering or various types of non-speech dysfluencies.

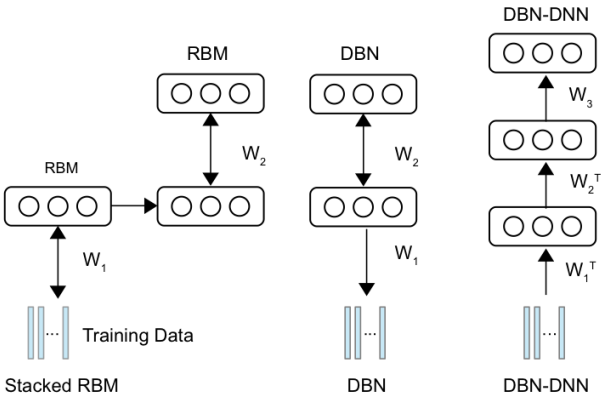

Figure 4: *DBN-DNN overview, after [13]*

3. Experiment 1: stuttering detection

We use two different partitioning schemes to compare results according to different categories of interest (Figure 5), namely generic-vs-individual speaker models (i.e., speaker-independent vs. speaker-dependent), and dysarthric-vs-non-dysarthric individuals. A total of 120 repetition stutters occurred across all 3115 recordings of dysarthric speech, and a total of 42 repetition stutters occurred across all 5641 recordings of non-dysarthric speech. The male and female dysarthric speakers with the most stutter occurrences were used for individual analysis; specifically, male dysarthric speaker M04 with 32 stutters, and female dysarthric speaker F03 with 22 stutters. Among the non-dysarthric speakers, there is no significant difference between males and females, so the non-dysarthric speaker with the most stutter occurrences was used in further analysis, namely male control MC04 with 16 stutters.

All training and testing data sets were divided in the same way – 70% of stutter occurrences were randomly assigned to training and paired with a random utterance without any stutter. By balancing training class sizes, we avoid the problem of overfitting to devolved majority classification. Testing data consisted of the remaining 30% of repetition stutters.

An empirical question is whether stutter detection is more or less difficult in dysarthric speech, compared to non-dysarthric speech. Table 3 shows the average error rates of detecting repetition stuttering using 5-fold cross validation with MFCC and LPCC features. Clearly, across all models, accuracy increases monotonically as additional context is added. We also note that we obtain state-of-the-art accuracy for dysarthric speaker F03 using 10 frames of surrounding context, which is comparable to Czyzewski *et al.*'s work with rough sets [4]. An n-way analysis of variance reveals strong effects of window size ($F_3 = 836.91, p < 0.001$) and population ($F_1 = 11.80, p < 0.01$), but not of the feature set ($F_1 = 0.12, p = 0.74$). Across all experiments, LPCCs give slightly lower error than MFCCs, on average (20.17% vs. 20.32%, respectively). Except for the (relatively inaccurate) case where no context frames are used, generic control models always give higher error than generic dysarthric models, by absolute differences of 2% to 2.35%. It is important to note that we only consider main effects of these grouping variables – given the different dimensionality of MFCC and LPCC, one cannot make direct *interaction* comparisons across these groups and context sizes simultaneously.

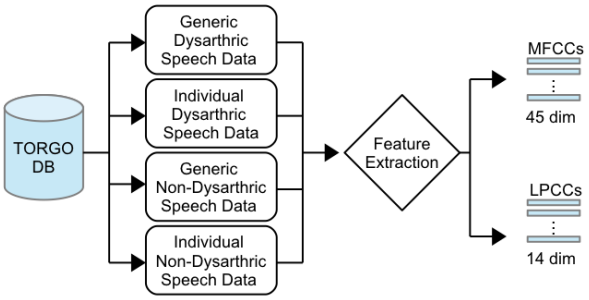

Figure 5: *Training & testing data set divisions used in experimentation.*

Speaker-dependent models always outperformed associated speaker-independent models. The difference in error rates between generic and individualized models is larger for dysarthric speech than non-dysarthric speech. At best, the speaker-dependent dysarthric models achieved a 5.06% lower rate than the speaker-independent dysarthric models, while speaker-dependent non-dysarthric models obtained at best a difference of 2.85%.

Interestingly, it is easier to detect stuttering in dysarthric speech than in non-dysarthric speech. In fact, error rates were consistently lower for the dysarthric speech (≈14%) than for the non-dysarthric speech (≈16%). This suggests that the implemented method is robust to this particular speech disorder.

4. Experiment 2: non-speech dysfluencies

We repeated the methodology of Experiment 1, but considered instead 'lower-level' dysfluencies and non-speech vocal noise that can affect speech recognition and synthesis systems.

Here, annotation is based on the phonetic transcriptions provided in the TORGO corpus. Segments labeled as *noi* (noise) were examined and manually tagged with either none, or any combination of the following three dysfluency types:

aspiration Noise related to breathing, i.e., inspiration or expiration.

mouth/lips Noise produced by the lips and/or mouth/tongue.

vocal Non-speech voicing (e.g., laughter, hesitation...).

Table 3: *Average error rate (%, 5-fold cross-validation) of stutter detection using MFCC and LPCC features across speaker groups. Speakers F03, M04, and MC04 are also examined individually due to their relatively high rates of stuttering.*

		Window size			
	Speaker(s)	0	±1	±3	±5
MFCC	F03	38.36	9.93	9.74	9.55
	M04	38.61	12.80	12.70	12.60
	all dysarthric	40.84	14.95	14.82	14.61
	MC04	38.24	14.49	14.27	14.05
	all controls	40.00	17.30	17.09	16.88
	all speakers	40.74	15.21	15.07	14.93
LPCC	F03	38.31	9.87	9.50	9.13
	M04	38.56	12.81	12.61	12.41
	all dysarthric	40.80	14.95	14.68	14.42
	MC04	38.18	14.44	14.00	13.57
	all controls	39.94	17.26	16.84	16.42
	all speakers	40.70	15.20	14.92	14.64

The procedure of classification and evaluation is the same as in Experiment 1, except only individuals with dysarthria are considered, since the amount of occurrences of such dysfluencies in control speakers were not significant. Among all 1403 recordings of the head-worn microphones for dysarthric speakers with phonetic transcriptions, we found 706 instances of aspiration noise, 496 of mouth/lips, and 111 of vocal noise.

Table 4: *Average error rate (%, 5-fold cross-validation) across other dysfluencies using MFCC and LPCC features across speaker groups.*

		Window size			
	Type	0	±1	±3	±5
MFCC	aspiration	39.98	19.19	19.60	19.11
	mouth/lips	43.28	24.95	24.81	24.68
	vocal	46.15	25.75	26.83	25.81
LPCC	aspiration	40.08	19.35	19.40	19.14
	mouth/lips	43.31	25.01	25.03	24.83
	vocal	46.18	25.81	25.92	25.42

Table 4 shows the average error rates of detecting the different non-speech dysfluencies using 5-fold cross validation with MFCC and LPCC features. The accuracy increases with the use of one or more frames of context, but adding more than one frame does not improve the results. These types of low-level dysfluencies are significantly localized in time or highly characterized by their spectral shape. Therefore, adding more contextual information does not appear to improve classification.

Dysfluencies of type *aspiration* are consistently more accurately classified than *mouth/lips*, which in turn are easier to classify than *vocal*. The *aspiration* dysfluencies contain a very characteristic timbre which is easier to discriminate from other speech sounds than the other classes. On the other hand, *vocal* dysfluencies are the closest to actual speech phones, leading to a more difficult differentiation. We note that *aspiration* dysfluencies are usually longer and since, in our current setting, an entire region is tagged with the noise type without performing segmentation, frames containing *aspiration* may be systematically more accurately labelled than those with other more localized noises such as *mouth/lips* or *vocal*.

5. Discussion and future work

We investigated the ability of a DBN-DNN to classify repetition stuttering and non-speech dysfluencies in dysarthric and non-dysarthric speech using MFCCs and LPCCs as input. Results indicate that repetition stuttering is detected with very similar (though significantly different) error rates across dysarthric and non-dysarthric speech. Increasing the dimension of the input, across either feature to the DBN-DNN consistently lowers the error rate, and there is no statistically significant difference between using MFCC or LPCC input features. Moreover, we find that among non-speech dysfluencies, aspiration is more accurately identified than mouth/lip dysfluency, which in turn is more accurately identified than other vocal activity. In both cases, a greater investigation into the effect of context is needed.

Overall, the results achieved here are comparable to similar work discussed in Section 1. However, given the somewhat limited number of stuttering and non-speech disturbances within TORGO, the results can be considered preliminary; more work with additional data sets would be needed to make more conclusive claims.

Since dysarthric speakers are more likely to stutter than non-dysarthric speakers, this must be considered when comparing across groups, especially when comparing aggregate speaker-independent models. Future work includes additional types of stuttering detection, including prolongations and stopgaps in spontaneous speech. We are also interested in extending and combining additional feature types, including autoencoders, and alternatives to the DBN structure itself. However, this paper has clearly shown that state-of-the-art stuttering detection, which had previously focused on non-pathological speech, can be applied to dysarthric speech. This automates a crucial component in systems that automatically improve the intelligibility of speech signals. Specifically, correcting dysfluencies has previously been shown to be a highly (if not the most) effective transformation that can be applied to speech signals [2]. Whereas that work depended on gold-standard phonemic transcriptions, our current work on stutters is relatively accurate given only the acoustics.

6. Acknowledgements

This work is partially funded by Thotra Incorporated, of which Frank Rudzicz is the CEO. It is also supported by an NSERC Discovery grant (RGPIN 435874) and a grant from the Nuance Foundation.

7. References

[1] A. B. Kain, J.-P. Hosom, X. Niu, J. P. H. van Santen, M. Fried-Oken, and J. Staehely, "Improving the intelligibility of dysarthric speech," *Speech Communication*, vol. 49, no. 2, pp. 743–759, 2007.

[2] F. Rudzicz, "Adjusting dysarthric speech signals to be more intelligible," *Computer Speech and Language*, vol. 27, no. 6, pp. 1163–1177, 2013.

[3] L. S. Chee, O. C. Ai, and S. Yaacob, "Overview of automatic stuttering recognition system," in *Proc. International Conference on Man-Machine Systems*, no. October, Batu Ferringhi, Penang Malaysia, 2009, pp. 1–6.

[4] A. Czyzewski, A. Kaczmarek, and B. Kostek, "Intelligent Processing of Stuttered Speech," *Journal of Intelligent Information Systems*, vol. 21, no. 2, pp. 143–171, 2003.

[5] M. Wiśniewski, W. Kuniszyk-Jóźkowiak, E. Smoka, and W. Suszyski, "Automatic detection of disorders in a continuous

speech with the hidden Markov models approach," in *Advances in Soft Computing*, 2007, vol. 45, pp. 445–453.

[6] ——, "Automatic detection of prolonged fricative phonemes with the hidden Markov models approach," *Journal of Medical Informatics & Technologies*, vol. 11, pp. 293–297, 2007.

[7] S. P. Rath, D. Povey, K. Vesel, and J. Cernock, "Improved feature processing for Deep Neural Networks," in *Intespeech*, 2013, pp. 109–113. [Online]. Available: http://www.danielpovey.com/files/2013_interspeech_nnet_lda.pdf

[8] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[9] L. S. Chee, O. C. Ai, M. Hariharan, and S. Yaacob, "Automatic detection of prolongations and repetitions using LPCC," *International Conference for Technical Postgraduates 2009, TECHPOS 2009*, 2009.

[10] U. Bhattacharjee, "A comparative study of LPCC and MFCC features for the recognition of Assamese phonemes," *International Journal of Engineering Research & Technology*, vol. 2, no. 3, pp. 1–6, 2013.

[11] T. Gulzar, A. Singh, and S. Sharma, "Comparative Analysis of LPCC , MFCC and BFCC for the Recognition of Hindi Words using Artificial Neural Networks," *International Journal of Computer Applications*, vol. 101, no. 12, pp. 22–27, 2014.

[12] F. Rudzicz, A. K. Namasivayam, and T. Wolff, "The TORGO database of acoustic and articulatory speech from speakers with dysarthria," *Language Resources and Evaluation*, vol. 46, no. 4, pp. 523–541, 2012.

[13] M. Tanaka and M. Okutomi, "A Novel Inference of a Restricted Boltzmann Machine," in *International COnference on Pattern Recognition*, 2014, pp. 1526–1531. [Online]. Available: http://www.ok.ctrl.titech.ac.jp/ mtanaka/ICPR2014mtanaka.pdf