Detecting Stuttering Events in Transcripts of Children's Speech

Sadeen Alharbi[1], Madina Hasan[2], Anthony J H Simons[3], Shelagh Brumfitt[4], and Phil Green[5]

Computer Science Department, The University of Sheffield,
Sheffield, United Kingdom.
{ssmalharbi1,m.hasan,a.j.simons,s.m.brumfitt,p.green}@sheffield.ac.uk

Abstract. Stuttering is a common problem in childhood that may persist into adulthood if not treated in early stages. Techniques from spoken language understanding may be applied to provide automated diagnosis of stuttering from children speech. The main challenges however lie in the lack of training data and the high dimensionality of this data. This study investigates the applicability of machine learning approaches for detecting stuttering events in transcripts. Two machine learning approaches were applied, namely HELM and CRF. The performance of these two approaches are compared, and the effect of data augmentation is examined in both approaches. Experimental results show that CRF outperforms HELM by 2.2% in the baseline experiments. Data augmentation helps improve systems performance, especially for rarely available events. In addition to the annotated augmented data, this study also adds annotated human transcriptions from real stuttered children's speech to help expand the research in this field.

Keywords: Stuttering event detection, Speech disorder, human-computer interaction, CRF, HELM

1 Introduction

Stuttering, sometimes referred to as 'stammering', is a speech disorder problem that starts in childhood and may result in severe emotional, communicational, educational and social maladjustment. Inadequate diagnoses and intervention at an early age may increase the risk that the condition may become chronic and has negative consequences on children with stuttering and their families [5, 2]. Thus, clinical intervention should take place as early as the preschool years because later intervention does not help. Also, it is not possible to determine a child's chance of naturally recovering from stuttering. Moreover, children are less tractable as they get older due to the reduction of neural plasticity [11]. During the assessment phase, clinicians need to carefully measure the stuttering events to determine the severity of stuttering. This measurement is usually conducted by counting the number of stuttering events in the child's speech. This process

is extremely dependent on the clinician's experience [1]. In another approach, the clinician transcribes a recorded session and classifies each spoken term into one of several normal, disfluent or stuttering categories [4]. This process takes a long time because of the need to write every spoken word which takes time and effort, requires knowledge of the relevant categories. An automated speech transcription of the recorded speech using Automatic Speech Recognition (ASR) could help clinicians speed up the assessment process and store the data for further investigations. However, understanding children's speech is well known to be a challenge even for humans, due to several factors, such as speech spontaneity, slow rate of speech and variability in the vocal effort [13]. Therefore, a large amount of data is required to train an ASR with an acceptable word error rate (WER) and to process the ASR output to automatically identify the stuttering events in the transcription.

Research in this area investigate three main approaches to detect stuttering events. The first area of study attempts to detect stuttering events from recorded speech signals. Howell and Sackin [9], for example, proposed the first attempt at stuttering recognition. Their study applied Artificial Neural Network (ANNs) and focused on identifying repetitions and prolongations. The basic idea is that the input vector of ANNs are the autocorrelation function and envelope. Their best accuracy was 80%. Geetha et al. [3] presented an objective method of differentiating stuttering disfluencies. They used ANN techniques on two groups of disfluent children. Several features were chosen to discriminate between normal and stuttering speech. They reported that ANN classifiers could predict the classifications of normal and stuttering with 92% accuracy. Another approach detects stuttering events from transcriptions. Mahesha and Vinod [15] is used a lexical Rule-Based (RB) algorithm to detect and estimate the severity of 4 types of stuttering events: Interjection (I), word repetition (W), sylable repetition (S) and prolongation (P), in orthographic transcripts from University College London's Archive of Stuttered Speech (UCLASS) [8]. In particular, they use prior domain knowledge to construct expert-based sets of rules to count the number of occurrences of each of the 4 stuttering events. The third approach is a combination of the previous two approaches. An automatic speech recognition approach has been proposed by Heeman et al [7, 6] in an attempt to merge a clinician's annotations with an ASR transcript to produce an annotated transcript of audio files (between 1 and 2 minutes duration) of read speech. Three types of stuttering were considered in [6]; revisions, interjections, and phrase, word and sound repetitions. However, the proposed system relied on the availability of the clinician's annotations of the read recordings.

This work investigates the detection of stuttering events in orthographic transcripts from UCLASS corpus. Traditional RB algorithm, for event detection tasks, is powerful in transferring the experiences of domain experts to make automated decisions. For offline applications where time and effort are not concerns and it can work with high accuracy for limited target data. However, this approach depends on the expert's knowledge [14], which means it only works if all situations of stuttering events are considered. This condition cannot be satis-

fied in practice due to the continuous variability in data volume and complexity. Moreover, this knowledge based approach is deterministic as it uses rules like "If word W is preceded by word Z, within C number of words, trigger the event Y", and if such scenarios are missed false decisions will be made without giving probability that evaluates those decisions.

Alternative probabilistic approaches are therefore required to learn the rules from the structure embedded in the data (i.e the stuttering pattern encapsulated in the stuttering sentences). Machine learning classifiers such as Hidden Event Language Model (HELM) and Conditional Random Fields (CRF) can actually help build data driven rules, and furthermore, as we find more data, these classifiers can be easily and frequently retrained. As a precursor to developing ASR for children with stuttering, this work investigates the applicability of machine learning approaches; particularly HELM and CRF, for automatically detecting stuttering events in transcripts of children's speech. Moreover, it is well known that the main limitation in children's speech related research is the lack of large publicly available corpora. To slightly alleviate the lack of training data in this field, additional recordings (from the children recordings in Release One of UCLASS has been transcribed and annotated with the stuttering events to support the research in this field. This study also examines the effect of augmenting the training data with artificially generated data. The rest of the paper is organised as follows. The guidelines and methodology used for producing the stuttering data transcriptions and annotations are described in Section 2. Section 3 presents the process of data normalisation and extraction of classification features. The two classification approaches are then described in Section 4. The data augmentation design and process is presented in Section 5. Section 6 explains the common measures used in stuttering events detection. Section 7 presents the experiments used in this study. Finally, the conclusion and future work are discussed in Section 8.

2 Data Transcription and Annotation

2.1 Data Transcription

This study uses the 31 publicly available orthographic transcriptions of children's speech monologue in Release One of UCLASS [8]. The transcription method in this release adopting certain conventional orthographies to indicate stuttering disfluencies. For example, "This is is a a a amazing". In addition to those transcriptions, this study adds the orthographic transcriptions of another 32 files from the same release following the same transcription guidelines. The data consists of 45 males and 18 females between 7 and 17 years of age. The 63 transcription files were then annotated to include the stuttering type for each word using the annotation approach described in Section 2.2.

2.2 Data Annotation Approach

The annotation approach followed in this study is the one proposed by Yairi and Ambrose [21] and used by Fabiola and Claudia [16]. In this approach, eight types

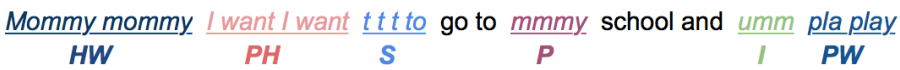

Fig. 1. Stuttering examples

of stuttering are considered: 1) sound repetitions, which include phoneme repetition (e.g., 'c c c complex'), 2) part-word repetitions, which consider a repetition of less than a word and more than a sound (e.g., 'com com complex'), 3) word repetitions that count the whole word repeated (e.g., 'mommy mommy'), 4) dysrhythmic prolongations, which involve an inappropriate duration of a phoneme sound (e.g., 'mmmmommy'), 5) phrase repetitions that repeat at least two complete words (e.g., 'this is this is'), 6) interjections, which involve the inclusion of meaningless words (e.g., 'ah', 'umm'), 7) revisions that attempt to fix grammar or pronunciation mistakes (e.g., 'I ate I prepared dinner'). 8) The block type includes inappropriate breaks in different parts of the sentence in between or within words. In this study, all types of stuttering were considered except the revision and block types. All stuttering types examined in the study are listed with their corresponding abbreviations in Table 1. Illustrative examples of the 6 different stuttering types are given in Figure 1. The annotation methodology was reviewed by a speech language pathologist (SLP), who is one of the co-authors[4] of this paper. The distribution of each type of stuttering event, as well as the number of words in the training and testing data, are summarised in Table 2.

Table 1. Stuttering types

Label	Stuttering Type
I	Interjection
S	Sound repetitions
PW	Part-word repetitions
W	Word repetitions
PH	Phrase repetitions
P	Prolongation
NS	Non Stutter

3 Data Normalisation and Features Extraction

Text normalisation is a very important step for the detection of stuttering events. It is also considered to be a prerequisite step for lots of downstream speech and language processing tasks. Text normalisation categorises text entities like dates, numbers, times and currency amounts, and transforms those entities into words. For our experiments, we normalised the transcriptions and extracted word level

Table 2. Data statistics

Set	Words	%I	%W	%PW	%S	%PH	%P	%NS
Train	11204	4.6	2.7	2.2	11.8	1.1	1.6	76
Test	2501	3.8	2.7	2.0	12.3	1.8	0.6	76.8
All Data	13705	4.5	2.6	2.1	11.9	1.2	1.4	76.3

based features to be used in the classification approaches used in this work. These features included n-grams for n = 2, 3 and 4, and up to two following words, referred to as post words.

4 Classification Approaches

4.1 Hidden Event Language Model

The Hidden Event Language Model (HELM) technique was adopted in this work, since it is an appropriate model to use when events of interest are not visible in every training context [17]. Stuttering events may be treated as hidden events, within a context that normally expects regular words. Standard language models are normally used to predict the next word and give word history. However, the language model here is applied to measure the probability of the appearance of each stuttering event at the end of each observed word, given its context. The inter-words events sequence are predicted by the model, E = e0, e1, e2,...en, based on given of a sequence of words, W =w0, w1, w2,...wn, using a quasi-HMM technique. The states of the model are represented as Word/event pairs, while the hidden state is represented as the stuttering event type. A standard language model provides the observations of previous words, and the probabilities.

4.2 Conditional Random Fields

Linear-Chain Conditional Random Fields (CRFs) are discriminative models that have been intensively used for sequence labelling and segmentation purposes [19]. The model aims to estimate and directly optimise the posterior probability of the label sequence, given a sequence of features (hence the frequently used term direct model). The CRF++ [12] toolkit was used in this work.

5 Data Augmentation

Data augmentation is a technique used for machine learning tasks in which there are too few training resources and usually not enough for training a model with reasonable performance. In speech processing, for example, the data augmentation is performed by adding perturbation from different sources such as artificial background noise, vocal tract length perturbation [10] and changing speaking

rate of the spoken utterance. For this study, we used a language model that was trained on the stuttering data (the training set), to generate additional sentences to supplement the original training data. The SRILM toolkit [18] was used to generate random sentences from a word-list, weighted by the probability of word-distribution in a language model. The word list was designed to include stuttering versions of the words in the publicly available word list (lm-csr-64k-vp-3gram) [20], in addition to the original word list.

The generated sentences (416,456 words) are of nonsense and not grammatically correct, most of the time, just like children's speech. Despite this fact, those generated sentences tend to exhibit feasible stuttering patterns, including less-common ones.

In order to automatically annotate the generated sentences, before it can be used for training the classifiers, an RB algorithm was built through several attempts with human annotators interventions. The annotation rules described in Section 2.2 were followed in this offline annotation process. A subset of 3000 words, was taken from the generated data and manually annotated as a reference. This reference was used to improve the performance of the RB algorithm. To further improve the labels on the generated data, some samples were revised and edited by human annotators. However, it is important to clarify that the RB annotation of the generated data is not fully revised by human annotators. Table 6 presents the labels distribution in the generated data.

6 Metrics

In this work, the conventional metrics: precision *Prec*, recall *Rec*, $F1$ score and accuracy *Acc* are used to evaluate the performance of the classifiers. The definitions of these metrics are given below.

$$\text{Prec} = \frac{TP}{TP + FP}, \qquad \text{Rec} = \frac{TP}{TP + FN}$$
$$\text{F1} = 2\frac{Precision * Recall}{Precision + Recall}, \text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}.$$

TP , FP and *FN* refer to true positive, false positive, and false negative counts, in that order.

7 Experiments

The following section presents our experiments on UCLASS data using the approaches discussed in Sections 4 and 5 for detecting stuttering events.

7.1 Baseline Experiments

Initial experiments were conducted to determine the best order of textual features to be used for training HELM and CRF classifiers. These initial experiments were performed using 10-fold cross-validation (CV) sets, to verify the

reliability of the model performance. Tables 3 and 4 show the CV results for HELM and CRF approaches, respectively. These results suggest that the best results of the HELM approach are obtained with 3-gram features, yielding an accuracy of 88%, Similarly, the best results for the CRF approach are obtained with 2-gram plus 2-post-words features with an accuracy of 90%. Generally, The CRF approach outperforms the HELM approach by **2.2%** relatively on accuracy.

Acceptable scores were obtained from both classifiers for detecting the HW, I and S classes. An important observation is the failure to detect the PH, PW, and P types of stuttering events. The main reason for this failure is referred to the scarcity of these classes in the data, as shown in Table 2. Based on these results, the rest of the experiments were designed to consider the 3-gram, and 2-gram plus 2-post-words features for HELM and CRF approaches, respectively. in order to avoid the cost of performing repeated cross-validation tests, we partitioned the data into training (80%) and evaluation (20%) sets and we deliberately ensured that the training and test sets had equal distributions of stuttering events from the start. Table 6 shows the distribution of the 6 different types of stuttering events in addition to the no stuttering event (NS). The initial experiments described above were also repeated on the defined training and evaluation sets, to check the generality of the defined sets. Table 5 shows the baseline results on the evaluation set. Similar observations to the cross-validation set of experiments are found.

Table 3. Cross-Validation results using HELM approach, with **Acc=90%**

N-gram	Stuttering-type	Precision	Recall	f1-score
2g	I	0.55	0.15	0.22
	W	0.99	0.88	0.93
	NS	0.86	0.99	0.92
	P	0.00	0.00	0.00
	PH	0.00	0.00	0.00
	PW	0.31	0.04	0.07
	S	0.92	0.65	0.76
3g	I	0.85	0.28	0.41
	W	0.99	0.82	0.90
	NS	0.87	1.00	0.93
	P	0.00	0.00	0.00
	PH	0.05	0.01	0.02
	PW	0.38	0.07	0.11
	S	0.96	0.65	0.78
4g	I	0.87	0.27	0.40
	W	0.99	0.80	0.88
	NS	0.87	0.99	0.93
	P	0.00	0.00	0.00
	PH	0.05	0.01	0.02
	PW	0.39	0.04	0.07
	S	0.96	0.67	0.78

Table 4. Cross-Validation results using CRF approach, with **Acc=92%**

N-gram	Stuttering-type	Precision	Recall	f1-score
2g+2p	I	0.78	0.23	0.34
	W	0.99	0.95	0.97
	NS	0.90	1.00	0.94
	P	0.00	0.00	0.00
	PH	0.20	0.02	0.03
	PW	0.25	0.04	0.07
	S	0.95	0.82	0.88
3g+2p	I	0.84	0.25	0.38
	W	0.99	0.95	0.97
	NS	0.90	0.99	0.94
	P	0.00	0.00	0.00
	PH	0.20	0.04	0.07
	PW	0.26	0.04	0.07
	S	0.95	0.82	0.88
4g+2p	I	0.91	0.21	0.34
	W	0.99	0.95	0.97
	NS	0.89	1.00	0.94
	P	0.00	0.00	0.00
	PH	0.10	0.03	0.04
	PW	0.33	0.05	0.08
	S	0.95	0.80	0.87

Table 5. HELM vs CRF results on the evaluation set, with **Acc=90%** and , **Acc=92%**, respectively

Classifier	Stuttering-type	Precision	Recall	f1-score
HELM	I	0.86	0.47	0.61
	W	0.96	0.85	0.90
	NS	0.89	0.99	0.94
	P	0.00	0.00	0.00
	PH	0.00	0.00	0.00
	PW	0.00	0.00	0.00
	S	0.98	0.78	0.87
CRF	I	0.89	0.35	0.50
	W	1.00	0.96	0.98
	NS	0.92	0.99	0.96
	P	0.00	0.00	0.00
	PH	0.00	0.00	0.00
	PW	0.00	0.00	0.00
	S	0.95	0.94	0.95

7.2 Effect of Data Augmentation

Using the technique explained in Section 5, 416,456 words were generated and annotated. The distributions of the 6 stuttering events in the generated data are presented in Table 6. The HELM and CRF models were retrained on the

Table 6. Data statistics of generated data

Words	%I	%W	%PW	%S	%PH	%P	%NS
416456	6.5	8.5	6.8	27.2	5.3	1.6	44.1

generated data, jointly with the original training data. The results of the retrained HELM and CRF classifiers on detecting and classifying the 6 stuttering and non-stuttering events, on the evaluation set, are presented in Table 7. Compared to the baseline results in Table 5, the performance of both classifiers was improved, with accuracies of 92%, and 94% for HELM and CRF approaches, respectively. These results also show that the performance of the CRF classifier was improved for all labels, including for those events that were infrequent in the original training data. The improvement obtained by the retrained HELM is however less, compared to that obtained by the CRF approach. Both classifiers still fail to detect the PH events. This is, however, expected due to the fact that the method used in the augmentation is based on a word list, not a list of phrases. Finally, despite the general improvements obtained by retraining using the augmented data, there is slight deterioration in the detection of NS, the dominant class, as shown in the CRF confusion matrix 8. This deterioration may due to the noisy labels of the generated data.

Table 7. Effect of data augmentation on the performance of HELM and CRF, when used to detect the stuttering events on the evaluation set

Classifier	Stuttering-type	Precision	Recall	f1-score
HELM	I	0.85	0.52	0.64
	W	0.97	0.74	0.84
	NS	0.91	0.99	0.95
	P	1.00	0.75	0.86
	PH	0.00	0.00	0.00
	PW	1.00	0.49	0.65
	S	0.92	0.84	0.88
CRF	I	0.96	0.49	0.65
	W	1.00	1.00	1.00
	NS	0.93	0.99	0.96
	P	1.00	0.57	0.73
	PH	0.00	0.00	0.00
	PW	0.61	0.32	0.42
	S	0.97	0.93	0.95

Table 8. CRF confusion matrix of stuttering event detection on the evaluation set: before and after augmentation

	Stuttering-type	I	W	NS	P	PH	PW	S
CRF trained on train set	I	30	0	50	0	0	3	4
	W	0	90	4	0	0	0	0
	NS	2	0	1910	1	0	0	7
	P	0	0	13	0	0	0	2
	PH	0	0	44	0	0	0	0
	PW	2	0	32	0	0	0	1
	S	0	0	19	0	0	0	284
	Stuttering-type	I	W	NS	P	PH	PW	S
CRF trained on aug-mented data	I	46	0	48	0	0	0	0
	W	0	94	0	0	0	0	0
	NS	0	0	1899	0	0	7	7
	P	0	0	5	8	0	0	1
	PH	0	0	44	0	0	0	0
	PW	2	0	21	0	0	11	0
	S	0	0	21	0	0	0	285

8 Conclusions and Future Work

In this work we studied the performance of HELM and CRF approaches as alternatives to the expert-based RB approach, in detecting the stuttering events in orthographic transcripts. Experimental results show that CRF consistently outperforms the HELM approach. Baseline experiments show how low frequency stuttering events ($PW/PH/P$) fail to be detected by both HELM and CRF classifiers, because those rare events were not seen or seen infrequently in the training set. In an attempt to increase the training data to improve the performance of these classifiers, data augmentation approach was adopted to generate additional random sentences according to an n-gram distribution pattern of words with probability of some stuttering event. Despite the fact that generated sentences are only probability-weighted nonsense, they tend to exhibit feasible stuttering patterns, including less common ones. Data augmentation helped improve the performance of both classifiers, especially for infrequent events. Experimental results reflect how the augmented data helped the CRF approach to improve the recovery of most labels including the rare P and PW events. However, PH events were still challenging to both classifiers. A phrase-based augmentation method, for sentence generation that creates realistic phrase repetition, could be a suitable solution.
Another contribution of this study has been to enlarge the corpus of human-transcribed stuttering speech data. We have approximately doubled the number of annotated sentences in the UCLASS corpus.

Acknowledgments. This research has been supported by the Saudi Ministry of Education, King Saud University

References

1. Brundage, S.B., Bothe, A.K., Lengeling, A.N., Evans, J.J.: Comparing judgments of stuttering made by students, clinicians, and highly experienced judges. Journal of Fluency Disorders 31(4), 271–283 (2006)
2. Craig, A., Calver, P.: Following up on treated stutterers studies of perceptions of fluency and job status. Journal of Speech, Language, and Hearing Research 34(2), 279–284 (1991)
3. Geetha, Y., Pratibha, K., Ashok, R., Ravindra, S.K.: Classification of childhood disfluencies using neural networks. Journal of fluency disorders 25(2), 99–117 (2000)
4. Gregory, H.H., Campbell, J.H., Gregory, C.B., Hill, D.G.: Stuttering therapy: Rationale and procedures. Allyn & Bacon (2003)
5. Hayhow, R., Cray, A.M., Enderby, P.: Stammering and therapy views of people who stammer. Journal of Fluency disorders 27(1), 1–17 (2002)
6. Heeman, P.A., Lunsford, R., McMillin, A., Yaruss, J.S.: Using clinician annotations to improve automatic speech recognition of stuttered speech. Interspeech 2016 pp. 2651–2655 (2016)
7. Heeman, P.A., McMillin, A., Yaruss, J.S.: Computer-assisted disfluency counts for stuttered speech. In: INTERSPEECH, pp. 3013–3016 (2011)
8. Howell, P., Davis, S., Bartrip, J.: The university college london archive of stuttered speech (uclass). Journal of Speech, Language, and Hearing Research 52(2), 556–569 (2009)
9. Howell, P., Sackin, S.: Automatic recognition of repetitions and prolongations in stuttered speech. In: Proceedings of the first World Congress on fluency disorders. vol. 2, pp. 372–374 (1995)
10. Jaitly, N., Hinton, G.E.: Vocal tract length perturbation (vtlp) improves speech recognition. In: Proc. ICML Workshop on Deep Learning for Audio, Speech and Language (2013)
11. Jones, M., Onslow, M., Packman, A., Williams, S., Ormond, T., Schwarz, I., Gebski, V.: Randomised controlled trial of the lidcombe programme of early stuttering intervention. BMJ 331(7518), 659 (2005), `http://www.bmj.com/content/331/7518/659`
12. Kudoh, T.: Crf++ (2007)
13. Liao, H., Pundak, G., Siohan, O., Carroll, M., Coccaro, N., Jiang, Q.M., Sainath, T.N., Senior, A., Beaufays, F., Bacchiani, M.: Large vocabulary automatic speech recognition for children. In: Interspeech (2015)
14. Liu, H., Gegov, A., Cocea, M.: Complexity control in rule based models for classification in machine learning context. In: UK Workshop on Computational Intelligence. vol. 513, pp. 125–143. Springer (2016)
15. Mahesha, P., Vinod, D.: Using orthographic transcripts for stuttering dysfluency recognition and severity estimation. In: Intelligent Computing, Communication and Devices, pp. 613–621. Springer (2015)
16. Staróbole Juste, F., Furquim de Andrade, C.R.: Speech disfluency types of fluent and stuttering individuals: age effects. Folia Phoniatrica et Logopaedica 63(2), 57–64 (2010)
17. Stolcke, A., Shriberg, E.: Statistical language modeling for speech disfluencies. In: Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on. vol. 1, pp. 405–408. IEEE (1996)
18. Stolcke, A., et al.: Srilm-an extensible language modeling toolkit. In: Interspeech. pp. 901–904 (2002)

19. Tseng, H., Chang, P., Andrew, G., Jurafsky, D., Manning, C.: A conditional random field word segmenter for sighan bakeoff 2005. In: Proceedings of the fourth SIGHAN workshop on Chinese language Processing. vol. 171. Citeseer (2005)
20. Vertanen, K.: Csr lm-1 language model training recipe (2007)
21. Yairi, E., Ambrose, N.G.: Early childhood stuttering ipersistency and recovery rates. Journal of Speech, Language, and Hearing Research 42(5), 1097–1112 (1999)