Comparing Detection Methods for Pause-Internal Particles

Mikey Elmers

Language Science and Technology, Saarland University, Saarbrücken, Germany
elmers@lst.uni-saarland.de

Abstract: This study investigates different machine learning architectures for classifying pause-internal phonetic particles (PINTs), such as filler particles (FPs), breath noises complementary to silences, and tongue clicks. Many of these PINTs co-occur, and by modeling them simultaneously, the aim is to improve the classification accuracy for the surrounding PINTs as well. An annotated subset from a German spontaneous speech corpus was used for modeling. Mel-frequency cepstral coefficients were used as inputs to model PINTs with three kinds of neural networks: a general neural network, a convolutional neural network, and a recurrent neural network. The models used the same hyperparameters, number of layers, and number of neurons for those layers, so that the focus was put onto the model architecture. The recurrent neural network was expected to perform the best since it is able to capture temporal information; however, all models performed similarly. The models performed best at classifying silent segments, followed by inhalations and exhalations. However, all models failed to accurately classify FPs and clicks, indicating that modeling PINTs simultaneously doesn't always improve accuracy for surrounding PINTs. These findings suggest that accurate classification is more dependent on annotation quantity and quality than model architecture. The main contributions of this paper are the classification of multiple PINTs simultaneously, and the improvement of PINTs classification for the German language.

1 Introduction

Pause-internal phonetic particles (PINTs) have a wide variety of functions and applications. For example, silent segments have an important role in breaking up speech. Silent segments refer to periods of acoustic-phonetic silence that are silent in production but not in transmission. Silent segments are defined similarly to the definition used by [1]. In other words, silent segments refer to a phase absent of phonetic particles such as breath noises, clicks, laughter, etc. Breaths frequently display a relationship with prosodic breaks and turn-taking. Filler particles (FPs) exhibit communicative functions for turn-taking and maintaining the floor [2], and as a sociolinguistic identifier [3]. Additionally, FPs also have technological applications for forensic voice comparison [4], and can improve text-to-speech (TTS) by reducing the cognitive load for the listener [5]. Examples of FPs in German are *äh* and *ähm* (*uh* and *uhm* in English).

The inclusion of PINTs in synthetic speech can improve naturalness and intelligibility. For synthetic speech, pauses have been shown to improve digit recollection [6], whereas breath noises improve sentence recollection [7]. The detection and modeling of breath groups can improve the quality of speech synthesis [8, 9]. Previous work [10] has indicated the importance of quality training data for TTS applications. Most modern TTS systems are unable to generate PINTs with appropriate location, duration, and frequency, especially for spontaneous conversational situations. Similar to [8], an additional goal of this work is to incorporate this detection

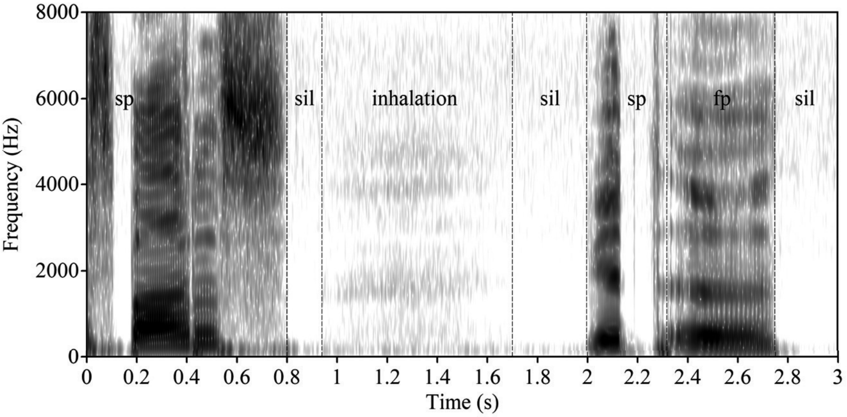

Figure 1 – Spectrogram example of annotated PINTs with speech (sp), silent segments (sil), inhalation, and filler particle (fp) taken from spontaneous speech.

method into a future TTS pipeline, for generating appropriate PINTs for spontaneous synthesis. These TTS systems can then be incorporated further into robotics, call centers, digital agents, etc.

Often PINTs co-occur with one another in a variety of sequences. For example, it is common to find sequences that contain multiple PINTs, such as an inhalation flanked by one or more silent segments (see Figure 1). Condron et al. [11] showed that training with more classes improved performance for non-verbal vocalizations (similar to PINTs) and laughter detection. The traditional approach has been to search for a single PINT, while collapsing all other PINTs to an 'other' class, or ignoring them altogether. Since these particles are not usually detected together, there is an absence of studies that incorporate state-of-the-art methods for detecting multiple PINTs simultaneously, especially for the German language. I expect that the classification of PINTs will benefit from simultaneous modeling, by training with multiple classes of PINTs, and have a positive outcome on synthesis quality for future research.

The rest of paper is structured as follows. Section 2 discusses the benefits of PINTs audio classification as well as some of the popular modeling techniques. Section 3 includes information regarding the corpus, data pre-processing, and model architectures. Section 4 provides results from the modeling work, and section 5 consists of a discussion and conclusion.

2 Related Work

There are many applications for audio classification including medical, automatic speech recognition (ASR), and TTS. Previous classification research has distinguished between coughs and breath noises [12], and detected respiratory disorders [13, 14]. Fukuda et al. [15] found a reduction in error rate when using breath events as a delimiter for ASR, and [9] found that annotating breath groups, and including breath noises, while omitting low probability breath events, created more fluent TTS.

Many methods have previously been used to detect PINTs: for silent segments [16, 17, 18], breath noises [8, 11, 16, 18], filler particles [19, 20, 21, 22], and clicks [11, 18]. Classification of PINTs have been done using a variety of methods, such as convolutional neural networks (CNN) [14], support vector machines (SVM) [18], Gaussian mixture models (GMM) [21], decision tree algorithms [23], and template matching [24, 25].

In a pilot study conducted with a small English dataset, a neural network (NN) was used to perform a binary classification, predicting breath noises using mel-frequency cepstral coefficients (MFCCs) as input. Historically, MFCCs have performed well for audio classification. This approach appeared promising for the task of locating PINTs. Machine learning algorithms are extremely prevalent in current research. This paper will model PINTs using a NN, a CNN, and a recurrent neural network (RNN). The RNN is expected to outperform the other models since it is able to evaluate the temporal relationship between different PINTs.

3 Methods

3.1 Corpus

The Pool corpus [26] consists of 100 male native speakers of German (age range 21–63 years old; mean age 39 years old). The present study considers the combination of the free technical setting with the spontaneous speech task, i.e. a picture description task. Similar to the board game Taboo, the speaker must describe a picture while not using any of the words listed beneath the picture.

This corpus has been annotated with information for different PINTs. There are 100 files in total (duration range 124–374 s; mean duration 223 s; total duration 6.2 hours). All signals are sampled at 16 kHz on a single channel. From these files, a total of 17,641 annotated PINTs were extracted (see Table 1). Additional classes were annotated like laughter, nasal filler particles (hm), glottal reflex, and other disfluencies like lengthening, truncation, and repair. However, their occurrences were too infrequent to include in the modeling.

Table 1 – Overview of annotated PINTs. Total refers to the durational total and prop is the individual PINTs durational total divided by the total time of the corpus.

class	count	min	max	mean	sd	total	prop
silent segment	10237	0.01 (s)	20.01 (s)	0.65 (s)	0.95 (s)	111.04 (min)	29.92%
inhalation	2891	0.05 (s)	2.10 (s)	0.51 (s)	0.27 (s)	24.79 (min)	6.68%
exhalation	1887	0.03 (s)	3.23 (s)	0.38 (s)	0.28 (s)	12.15 (min)	3.27%
filler (uh)	1156	0.04 (s)	1.44 (s)	0.35 (s)	0.16 (s)	6.81 (min)	1.83%
filler (uhm)	549	0.15 (s)	2.64 (s)	0.53 (s)	0.25 (s)	4.85 (min)	1.30%
click	921	0.00 (s)	0.50 (s)	0.06 (s)	0.05 (s)	0.96 (min)	0.25%

3.2 Data Pre-processing

The first step for pre-processing was to extract 13 MFCCs with a frame size of 93 ms, and a hop length of 23 ms, using the *Librosa* python package [27]. Where the files differed in duration zero-padding was used in order to maintain the same size for modeling. The models were trained on the following nine classes: silent segments, inhalation, exhalation, two FPs ("uh" and "uhm"), clicks, task change (long stretches of silence while the interviewer changes tasks), zero-padding, and a final category for speech.

3.3 Model Architecture and Training

Models were implemented using *Keras* [28]. All models are compiled using a sparse categorical cross entropy loss function, a learning rate of 0.0001, the Adam optimizer, a batch size of 32, and for 40 epochs. A training/test split of 75/25 is used for all the models. Additionally, 20%

of the training set is used for validation. Since there are 100 files, 60 files of material were randomly selected for training, 15 files of material were randomly selected for validation during model training, and 25 files of material were randomly selected and withheld for testing. Each model was trained using a different training/test split.

3.4 Neural Network

The NN model incorporates a flattened input layer followed by two fully connected hidden layers, each with 64 neurons, a rectified linear unit (ReLU) activation function, and a 30% dropout for each layer. The output is a softmax layer to predict the output class. Training time is approximately 25 minutes on CPU.

3.5 Convolutional Neural Network

The CNN model is comprised of two 1D convolutional layers. Each with 32 filters (size = 1, stride = 1), a ReLU activation function, followed by a 1D max pooling and batch normalization. The output is then flattened and fed into a dense layer with 64 neurons and a ReLU activation function, with a dropout of 30% applied to this layer. The output is a softmax layer for predicting the output class. Training time is approximately 35 minutes on CPU.

3.6 Recurrent Neural Network

The RNN model consists of two fully connected long short-term memory (LSTM) layers each with 64 neurons. Next is a dense layer with 64 neurons, a ReLU activation function, and a 30% dropout. The output is a softmax layer which predicts the output class. Training time is approximately 70 minutes on CPU.

4 Results

Table 2 compares the accuracy, precision, recall, and F1 score for the three models. Both the CNN and the RNN performed slightly better than the NN in terms of accuracy and F1 score. The CNN and RNN performed similarly, except that the RNN performed better for precision. All models began with a relatively high accuracy and improved minimally throughout the remaining epochs. Overall, the scores for precision, recall, and F1 were lower than expected. Therefore, a confusion matrix was generated for each model (see Table 3) to further investigate the classification of individual PINTs. All three models performed best when classifying silent segments, followed by inhalations and exhalations. For both inhalations and exhalations, they were most often confused for a silent segment in all models. Overall, the models performed well when separating inhalations from exhalations and vice versa. However, all models failed to classify FPs and clicks. Table 4 compares model performance for the individual PINTs. The CNN performed best when classifying silent segments, the NN performed best when classifying inhalations, and both the CNN and RNN performed equally well for classifying exhalations.

Table 2 – Accuracy, Precision, Recall, and F1 Score for different models.

Model	Accuracy	Precision	Recall	F1 Score
NN	85.6%	53.5%	41.6%	40.5%
CNN	86.1%	53.2%	41.9%	41.8%
RNN	86.1%	69.0%	42.1%	41.7%

Table 3 – NN, CNN, and RNN confusion matrices for their respective test set. Rows correspond to annotated class and columns correspond to prediction.

NN							
class	**sil**	**inh**	**exh**	**uh**	**uhm**	**click**	**sum**
silent segment (sil)	64971	2743	789	-	-	-	68503
inhalation	4141	10372	58	-	-	-	14571
exhalation	3215	497	2188	-	-	-	5900
filler (uh)	60	3	34	-	-	-	97
filler (uhm)	68	4	33	-	-	-	105
click	209	85	6	-	-	1	301
sum	72664	13704	3108	-	-	1	89477

CNN							
class	**sil**	**inh**	**exh**	**uh**	**uhm**	**click**	**sum**
silent segment (sil)	66494	1375	754	-	-	1	68624
inhalation	5111	9351	100	-	-	-	14562
exhalation	3173	336	2532	-	-	-	6041
filler (uh)	53	2	27	-	-	-	82
filler (uhm)	80	5	20	-	11	-	116
click	181	73	11	-	-	-	265
sum	75092	11142	3444	-	11	1	89690

RNN							
class	**sil**	**inh**	**exh**	**uh**	**uhm**	**click**	**sum**
silent segment (sil)	64771	1813	811	-	-	-	67395
inhalation	4214	10098	113	-	-	-	14425
exhalation	2812	394	2308	-	-	-	5514
filler (uh)	38	2	13	-	-	-	53
filler (uhm)	50	2	17	-	3	-	72
click	165	74	8	-	-	3	250
sum	72050	12383	3270	-	3	3	87709

5 Discussion and Conclusion

This paper considered different machine learning architectures for classifying PINTs. Surprisingly, the NN, CNN, and RNN performed similarly, with some individual advantages in different cases. I had hypothesized that the RNN would perform better than the other two models since it is better able to capture temporal information. However, this was not the case. The models were able to easily identify silent segments and could classify inhalations fairly well, most likely due to them being the most frequently annotated classes. The models had middling success when attempting to detect exhalations. This is possibly due to the lower frequency of occurrence of exhalation annotations in the data. Inhalations and exhalations were sometimes misclassified as silent segments, possibly due to their frequent proximity.

Table 4 – Proportion correct for each model and class.

Model	sil	inh	exh	uh	uhm	click
NN	94.8%	71.2%	31.1%	0.0%	0.0%	0.3%
CNN	96.9%	64.2%	41.9%	0.0%	9.5%	0.0%
RNN	96.1%	70.0%	41.9%	0.0%	4.2%	1.2%

All models were unable to accurately classify FPs and clicks. This finding is counter to the hypothesis that modeling multiple PINTs simultaneously would improve the classification accuracy of other PINTs. The models might have had difficulty classifying FPs because they were too similar to the speech category. The models struggled to properly classify clicks, which were often incorrectly classified as a silent segment. This could be in part due to the extremely short duration of clicks or a drawback of using only MFCCs as input.

The models were designed to encourage parity between them by having a similar number of layers, neurons for those layers, and the same hyperparameters. This decision was made to highlight the architectural differences of the models. During training time all three models started with a relatively high accuracy and only improved slightly during subsequent epochs. Since all the models performed similarly, I hypothesize that further improvements in accuracy could be gained by increasing the number of occurrences for the PINTs, especially the infrequent ones, showcasing the importance of quality annotations. In addition to MFCCs, other acoustic features should be investigated in order to improve classification. Lastly, since the inputs were MFCCs, the CNN model used 1D convolutional layers. Classification could possibly be further improved by using spectrogram images instead of MFCCs for models using a CNN architecture.

A primary goal for developing these classification models is to improve speech synthesis. Future work will implement a PINTs classification method as part of the training process for a TTS pipeline to create more natural, conversational speech synthesis.

6 Acknowledgements

This research was funded in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID MO 597/10-1.

References

[1] Belz, M. and J. Trouvain: *Are 'silent'pauses always silent?* In *19. International Congress of Phonetic Sciences (ICPhS)*. 2019.

[2] Clark, H. H. and J. E. F. Tree: *Using uh and um in spontaneous speaking*. *Cognition*, 84(1), pp. 73–111, 2002.

[3] Fruehwald, J.: *Filled pause choice as a sociolinguistic variable*. In *New Ways of Analyzing Variation (NWAV44)*, pp. 41–49. Penn Graduate Linguistics Society, 2016.

[4] Hughes, V., P. Foulkes, and S. Wood: *Strength of forensic voice comparison evidence from the acoustics of filled pauses*. *International Journal of Speech, Language and the Law*, pp. 99–132, 2016.

[5] Dall, R., M. Tomalin, and M. Wester: *Synthesising filled pauses: Representation and datamixing*. In *9th ISCA Speech Synthesis Workshop*, pp. 7–13. 2016.

[6] Elmers, M., R. Werner, B. Muhlack, B. Möbius, and J. Trouvain: *Evaluating the effect of pauses on number recollection in synthesized speech*. In *Elektronische Sprachsignalverarbeitung 2021, Tagungsband der 32. Konferenz*, Studientexte zur Sprachkommunikation, pp. 289–295. TUD Press, Berlin, 2021.

[7] Elmers, M., R. Werner, B. Muhlack, B. Möbius, and J. Trouvain: *Take a breath: Respiratory sounds improve recollection in synthetic speech*. In *Proc. Interspeech 2021*, pp. 3196–3200. 2021. doi:10.21437/Interspeech.2021-1496.

[8] Székely, É., G. E. Henter, and J. Gustafson: *Casting to corpus: Segmenting and selecting spontaneous dialogue for tts with a cnn-lstm speaker-dependent breath detector*. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6925–6929. IEEE, 2019.

[9] Székely, É., G. E. Henter, J. Beskow, and J. Gustafson: *Breathing and speech planning in spontaneous speech synthesis*. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7649–7653. IEEE, 2020.

[10] Henter, G. E., S. Ronanki, O. Watts, M. Wester, Z. Wu, and S. King: *Robust tts duration modelling using dnns*. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5130–5134. IEEE, 2016.

[11] Condron, S., G. Clarke, A. Klementiev, D. Morse-Kopp, J. Parry, and D. Palaz: *Non-verbal vocalisation and laughter detection using sequence-to-sequence models and multi-label training*. In *Proc. Interspeech 2021*, pp. 2506–2510. 2021. doi:10.21437/Interspeech.2021-1159.

[12] Coppock, H., A. Gaskell, P. Tzirakis, A. Baird, L. Jones, and B. Schuller: *End-to-end convolutional neural network enables covid-19 detection from breath and cough audio: a pilot study*. *BMJ Innovations*, 7(2), 2021.

[13] Lei, B., S. A. Rahman, and I. Song: *Content-based classification of breath sound with enhanced features*. *Neurocomputing*, 141, pp. 139–147, 2014.

[14] Saraiva, A. A., D. Santos, A. Francisco, J. V. M. Sousa, N. M. F. Ferreira, S. Soares, and A. Valente: *Classification of respiratory sounds with convolutional neural network*. In *BIOINFORMATICS*, pp. 138–144. 2020.

[15] Fukuda, T., O. Ichikawa, and M. Nishimura: *Detecting breathing sounds in realistic japanese telephone conversations and its application to automatic speech recognition*. *Speech Communication*, 98, pp. 95–103, 2018.

[16] Braunschweiler, N. and L. Chen: *Automatic detection of inhalation breath pauses for improved pause modelling in hmm-tts*. In *8th ISCA Speech Synthesis Workshop*. 2013.

[17] Singh, L. G., N. Adiga, B. Sharma, S. R. Singh, and S. Prasanna: *Automatic pause marking for speech synthesis*. In *TENCON 2017 IEEE Region 10 Conference*, pp. 1790–1794. IEEE, 2017.

[18] Garcia, A., M. Collery, V. Miloulis, and Z. Malisz: *Classification and clustering of clicks, breathing and silences within speech pauses*. In *Proc. 5th Laughter Workshop*, pp. 6–9. 2018.

[19] Goto, M., K. Itou, and S. Hayamizu: *A real-time filled pause detection system for spontaneous speech recognition*. In *6th European Conference on Speech Communication and Technology*. 1999.

[20] Audhkhasi, K., K. Kandhway, O. D. Deshmukh, and A. Verma: *Formant-based technique for automatic filled-pause detection in spontaneous spoken english*. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4857–4860. IEEE, 2009.

[21] KRIKKE, T. F. and K. P. TRUONG: *Detection of nonverbal vocalizations using gaussian mixture models: looking for fillers and laughter in conversational speech.* In *Proc. Interspeech 2013*, pp. 163–167. 2013.

[22] REICHEL, U. D., B. WEISS, and T. MICHAEL: *Filled pause detection by prosodic discontinuity features. Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*, pp. 272–279, 2019.

[23] GERMESIN, S., T. BECKER, and P. POLLER: *Domain-specific classification methods for disfluency detection.* In *9th Annual Conference of the International Speech Communication Association*. 2008.

[24] RUINSKIY, D. and Y. LAVNER: *An effective algorithm for automatic detection and exact demarcation of breath sounds in speech and song signals. IEEE Transactions on Audio, Speech, and Language Processing*, 15(3), pp. 838–850, 2007.

[25] LU, L., L. LIU, M. J. HUSSAIN, and Y. LIU: *I sense you by breath: Speaker recognition via breath biometrics. IEEE Transactions on Dependable and Secure Computing*, 17(2), pp. 306–319, 2017.

[26] JESSEN, M., O. KÖSTER, and S. GFROERER: *Influence of vocal effort on average and variability of fundamental frequency. International Journal of Speech Language and the Law*, 12(2), pp. 174–213, 2005.

[27] MCFEE, B., C. RAFFEL, D. LIANG, D. P. ELLIS, M. MCVICAR, E. BATTENBERG, and O. NIETO: *librosa: Audio and music signal analysis in python.* In *Proceedings of the 14th python in science conference*, vol. 8, pp. 18–25. Citeseer, 2015.

[28] CHOLLET, F. ET AL.: *Keras.* `https://keras.io`, 2015.