INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Pause prediction from text for speech synthesis with user-definable pause insertion likelihood threshold

Norbert Braunschweiler, Ranniery Maia

Toshiba Research Europe Ltd., Cambridge Research Laboratory, Cambridge, United Kingdom
{norbert.braunschweiler,ranniery.maia}@crl.toshiba.co.uk

Abstract

Predicting the location of pauses from text is an important aspect for speech synthesizers. The accuracy of pause prediction can significantly influence both naturalness and intelligibility. Pauses which help listeners to better parse the synthesized speech into meaningful units are deemed to increase naturalness and intelligibility ratings, while pauses in unexpected or incorrect locations can reduce these ratings and cause confusion. This paper presents a multi-stage pause prediction approach including first prosodic chunk prediction, followed by a feature scoring algorithm and finally a pause sequence evaluation module. Preference tests showed that the new method outperformed a pauses-at-punctuation baseline while not yet matching human performance. In addition, the approach includes two more functionalities: (1) a user-specifiable pause insertion rate and (2) multiple output formats in the form of binary pauses, multi-level pauses or as a score reflecting pause strength.

Index Terms: pause prediction, phrasing, prosody, speech synthesis, machine learning

1. Introduction

Predicting pauses from text is an essential part in a text-to-speech (TTS) system. The presence of pauses supports listeners in parsing the speech stream and enables them to better digest the incoming information. Pauses contribute to signal meaningful, coherent units and increase intelligibility. On the other hand, well-timed pauses add to the naturalness impression of a speech synthesizer and can also induce expressiveness.

TTS systems typically predict pauses by extracting a number of features from input text. These features often include punctuation markers, which are sometimes used as fall-back strategy to avoid inserting pauses at linguistically unmotivated locations. A sentence can include a multitude of linguistically motivated pause locations but also a number of linguistically unacceptable ones. An example for an unacceptable pause insertion in the sentence *We learn something every day, and lots of times it's that what we learned the day before was wrong.* would be to insert a pause between *day* and *before*, because both words form a semantic unit in this sentence and it's unreasonable to break this unit by a pause. Therefore, the pause prediction task becomes one of minimizing the number of unacceptable pause insertions and maximizing the number of reasonable pause insertions while still maintaining a natural speech rhythm.

The prediction of pauses or prosodic breaks from text is a wide-studied topic in the field of TTS research. Methods range from rule based approaches [1, 2, 3] to a variety of machine learning methods like decision trees [4], HMMs [5, 6], Conditional Random Fields (CRFs) [7], Bidirectional Recurrent Neural Networks (BiRNNs) [8], and also combinations of methods [9, 10, 11, 12, 13]. There are many links into other fields such as linguistics, syntax, psychology and automatic speech recognition which include transcription systems for intonational phrase structure (i.e. the ToBI annotation method [14]), the relation between syntactic structure and pause insertion [15, 16], psycholinguistic studies related to the effects of superfluous and missing prosodic breaks [17, 18], and the recovery of punctuation in automatic speech recognition output for improved performance of spoken language translation systems [19, 20].

Related work published by Parlikar et al. [21] introduces the notion of a "knob" to change the number of phrase breaks their model produces. However, their implementation is fundamentally different than the approach chosen here because they are combining multiple phrasing models into a log-linear framework. Also, Mishra et al. [22] present an intonational phrase break prediction method which focusses on syntactic features to overcome the problems of linguistic rules and data-driven models not being able to generalize across domains. The approach presented here does also generalize across text styles by using syntactic features, but it uses additional and also non-syntactic features for pause prediction and a fundamentally different multi-stage approach. Mishra et al. [22] also address the task of the break predictor to be suitable for an incremental speech synthesizer, which is not considered in the current work.

Since pause prediction methods are known to produce a certain rate of linguistically unmotivated pauses, one strategy to avoid these is to restrict pauses to words which are marked by punctuation, i.e. commas, colons, etc. Therefore, this approach is used as a baseline method. The pauses-at-punctuation-only method almost certainly ensures the absence of bad pause insertions and text normalization typically prevents pauses in cases like hyphenated words or abbreviations such as "e.g.". However, it also under-predicts pauses, especially in longer sentences (e.g. $>$ 8 words) and the synthesis can sound unnatural, lacking natural phrasing. This will reduce the perceived naturalness of TTS and will also increase the cognitive effort needed by listeners to parse and understand the content of synthetic speech. The goal of the proposed method was to improve that situation. It builds upon previous work implemented in Toshiba's ToSpeak TTS system. Some of this work forms the initial part of the current approach, but the current approach evaluates and further refines the output by using a scoring system and sequence evaluation module. The new approach also has added functionalities, i.e. a user-definable threshold to specify the rate of pause insertions and multiple output formats in binary or graded form reflecting different levels of pause strength. This paper is structured as follows: first the architecture of the new pause prediction method is explained, this is followed by both objective and subjective evaluation of the new model, a discussion and finally conclusions.

http://dx.doi.org/10.21437/Interspeech.2016-752

Table 1: *Features used in pause predictor PauLo.*

Feature	Description
chunk	prosodic break value (3 word window)
orth	orthography of word (3 word window)
punc	punctuation of word (3 word window)
pos	part-of-speech of word (3 word window)
role	syntactic role of word (3 word window)
dist	dist. in words to head (3 word window)
numWds	number of words since start of sent

2. Method

The new pause prediction method, henceforth called **PauLo** for **Pau**se **Lo**cator', uses a multi-stage approach to predict pauses from text. Initial text processing is provided by Toshiba's ToSpeak TTS system including text analysis, text normalisation, feature generation, and prosodic chunk prediction. Input text is split into tokens, part-of-speech tagged (decision tree based tagger using C4.5 [23]), parsed (probabilistic left-to-right parser), normalized (expansion of digits, abbreviations etc.) and prosodic chunk boundaries are predicted. The prosodic chunk prediction module is based on previous work by Burrows et al. [24]. It uses a decision tree model (C4.5 with sub-setting [23]) trained on an American English TTS corpus which had been hand labelled with ToBI break indices. For the prosodic chunk predictor ToBI break levels 3 and 4 were merged into a single break level because this resulted in better performance when used in a pause prediction task in [24]. Therefore, it predicts the presence or absence of a prosodic break for each word juncture in a given sentence.

PauLo then enters prosodic chunks and further features, including orthography, punctuation, part-of-speech (POS) tags, syntactic role, distance in number of words to syntactic head, and position of word in sentence into a scoring algorithm. Table 1 shows the features used in PauLo which are mostly generated for a 3-word window. The scoring algorithm is designed to score the relative importance of features or feature combinations towards the presence of a pause. It generates a score on a scale from 0 – 100 by using a set of rules defining scores for individual features (e.g. punctuation is given a score of 100) or feature combinations. Scoring values were chosen on a development set including observations from many speech corpora and rules are designed to add or subtract from the score. Observations in the development set were taking frequency of occurrence of a certain feature or feature combination into account. Finally a module which re-evaluates the sequence of predicted pauses given a certain threshold is applied and may delete or insert pauses to generate the final pause prediction. The final pause sequence evaluation was included to avoid consecutive pause insertions and to increase pause insertion scores in sequences of words larger than 6 which do not have a score above a given threshold.

While the second stage is influenced by predictions in the first stage it is not restricted to only insert pauses at locations of predicted prosodic chunk boundaries.

Figure 1 shows a flowchart of the pause prediction method presented in this paper starting from plain text input to the output, i.e. the predicted pauses.

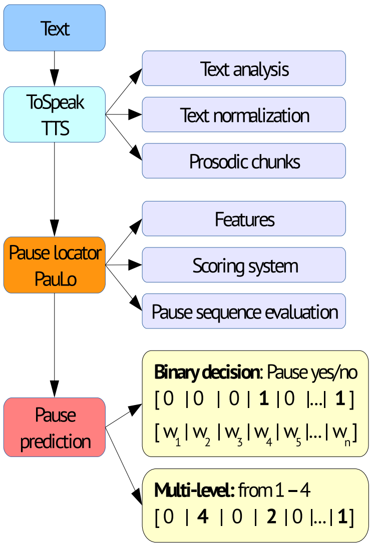

Figure 1: *Flowchart of modules from text to pause.*

Table 2: *Example output of PauLo.*

Word	Pauses	
	Binary	Levels
A	0	0
fence	1	2
cuts	0	0
through	0	4
the	0	0
corner	0	0
lot	1	1

A user can select the pause insertion rate by choosing a value from 0 – 100, where "0" means the lowest threshold resulting in the highest pause insertion rate and "100" representing the highest threshold resulting in the lowest number of pauses.

A user can also select the output mode, which by default generates a binary decision (pause yes or no), but PauLo can also produce a multi-level pause prediction (levels 1 – 4) as well as providing a score ranging from 0 – 100. Multi-level pause output is designed to represent different levels of pause strength. For this, the pause score is translated via threshold values into currently 4 levels, where "1" means the strongest pause and "4" the weakest pause. The decision to chose 4 levels was influenced by the existence of 4 break indices in the ToBI model [14]. However, the ToBI version is designed to mark the subjective strength of the association between one word and the next word and the scale is reversed to the current scheme, i.e. "0" for the strongest conjoining and "4" for the most disjoint.

Table 2 shows an example output of the pause predictor for one of the Harvard sentences [25] for both binary (pause locations indicated by 1) and multi-level pauses (pause locations and levels indicated by 1 – 4).

3. Evaluation

Objective evaluation was performed using an American English TTS corpus read by a professional female voice talent. It contained 2550 sentences with 28503 words and 3636 manually annotated pauses. The ratio of words followed by pauses to words not followed by pauses was 14.62%. Sentence final words were excluded because they have a pause attached by default.

The evaluation was performed by calculating the F-measure shown in equation (1). The F-measure considers both precision (2) and recall (3), and is therefore considered to be a more representative measurement for pause prediction performance given the skewed distribution of words followed by pauses to words not followed by pauses, which is typically in the range of 5 – 15% in a standard TTS corpus.

$$\mathbf{F} = 2 \times \frac{precision \times recall}{(precision + recall)} \tag{1}$$

$$\mathbf{P} = \frac{True\,positives}{(True\,positives + False\,positives)} \tag{2}$$

$$\mathbf{R} = \frac{True\,positives}{(True\,positives + False\,negatives)} \tag{3}$$

As baseline a pauses-after-punctuation only method was used (henceforth called: Punc). Method Punc inserted pauses at each punctuation marks, including comma, colon, semi-colon, hyphen and quotation marks.

Table 3 shows the results of the objective evaluation comparing pause insertion methods Punc and PauLo. For PauLo, the user-specifiable pause insertion likelihood threshold was set to 95, which was determined by measuring the optimal threshold value on a set of 501 test sentences from another American English TTS corpus. Figure 2 shows a plot of the changes in F-score as a function of changing the threshold between 1 – 100. As can be seen, for the 501 evaluation sentences the best threshold was at 95. The results show that the Punc method achieves the highest precision (reflecting the high overlap of punctuation and pauses) but has the lowest recall resulting in the lowest F-score overall. The best F-score is produced by PauLo which albeit showing lower precision than the baseline has a higher recall and overall receiving the highest F-score.

Table 3: *Comparison of pause insertion methods.*

	Punc	PauLo
Precision	0.8809	0.6984
Recall	0.3174	0.4637
F-score	0.4666	0.5574

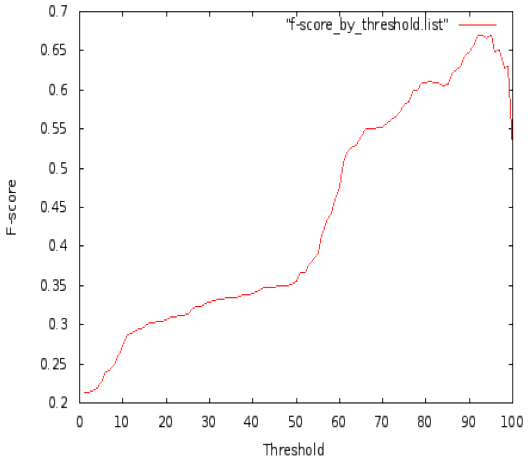

Figure 2: *F-score as function of threshold changes from 1 - 100.*

However, the F-score is not necessarily the best measure to predict the perceived performance of a pause predictor for TTS. Especially when it comes to the task of inserting more pauses in longer sentences. Here, a higher rate of pause insertions might result in a lower F-score, because there are more pauses in locations which do not overlap with locations from the speaker, but nevertheless are linguistically reasonable pause insertions. One of the methods to check this is to run a listening test and ask human subjects which method they prefer. Therefore, a subjective evaluation was performed by comparing synthesized speech samples including different pause locations from 4 different methods:

- Punc = pauses at punctuation
- PauLo = pauses predicted by PauLo
- Hand = pauses manually inserted by first author
- Spkr = pauses inserted by TTS speaker

Systems Hand and Spkr were added to measure PauLo's performance against pauses inserted by humans. System Spkr had pauses as inserted by the professional voice talent of the TTS source corpus. System Hand had pauses manually inserted by the first author and was intended to represent another way of possible, human-inserted pauses. Both systems were considered to provide an upper bound of possible pause insertion accuracy while at the same time showing the impact of variation in pause numbers and locations as well.

Evaluation samples were synthesized with an American English DNN-TTS whose female source speaker was different than the one used in the objective evaluation. The corpus included 4418 utterances for training and 100 for cross-validation. More than 500 sentences were set aside for testing. Audio stimuli were sampled at 22050 Hz. The DNN-TTS was used to synthesize stimuli which just differed in their number and position of pauses as provided by the pause insertion methods. Pause durations were taken as predicted by the synthesizer.

A series of preference listening tests was conducted. Subjects were asked which of two stimuli they preferred in terms of naturalness and had the option to say "neither". 6 subjects participated in the test. To compare all systems with each other 6 contrasts had to be considered. For each system contrast 15 evaluation sentences were selected, differing at least in one pause position/number. This requirement avoided identical sentences being used in the test.

Table 4: *Number of pauses in evaluation sentences.*

Comparison			# pauses		
PauLo	:	Punc	36	:	10
PauLo	:	Hand	27	:	47
PauLo	:	Spkr	33	:	24
Hand	:	Spkr	46	:	22
Hand	:	Punc	43	:	11
Sprk	:	Punc	31	:	11

Sentence selection was based on the set of test sentences from the TTS corpus. Because of the selection criteria mentioned above, individual system contrasts did not always include identical sentences but some differed. Test sentences covered a wide range of text styles including, news, navigation, questions, exclamations, and long sentences from literary works. Since the goal was to evaluate pause prediction performance, the test included pre-dominantly longer sentences, i.e. the number of words in sentences ranging from 5 – 32, with an average of 19.1 words per sentence (counted after text normalization). Table 4 shows the number of pauses in the 15 sentences selected for each contrast. System Punc has on average the lowest number of pauses, followed by Spkr, PauLo and Hand. PauLo used a pause insertion threshold of 95 which was chosen by comparing PauLo pause predictions at different thresholds against pauses inserted by the TTS source speaker in the 501 sentence test set as described above, i.e. see Figure 2.

The results of the preference tests are presented in Table 5. PauLo was significantly preferred over the Punc baseline but is still inferior to human inserted pauses, i.e. both Hand (albeit showing a non-significant difference) and Spkr. The increased number of pauses inserted in reasonable locations by PauLo was preferred by listeners on average compared to the much smaller number of pauses inserted by the Punc method. Interestingly, the larger number of pauses in system Hand compared to PauLo was tendentially preferred, but there is no statistical significance. System Spkr, on the other hand, can be considered to represent the upper limit of pause insertion performance since it was comprised by the pauses of the professional speaker of the TTS corpus. This hypothesis was confirmed by a significant preference to both Punc and PauLo, but not to system Hand which can be considered as just another reasonable alternative way to insert pauses into the given sentences.

Table 5: *Results of preference tests.*

PauLo	Punc	Neither	p-value
63.0%	23.9%	13.0%	0.004*
PauLo	Hand		
31.1%	46.6%	22.2%	0.322
PauLo	Spkr		
28.8%	53.3%	17.8%	0.046*
Hand	Spkr		
40.0%	48.8%	11.1%	0.349
Hand	Punc		
60.0%	21.1%	18.8%	<0.001*
Spkr	Punc		
79.8%	6.7%	13.5%	<0.001*

4. Discussion

The reason why the new pause prediction method was significantly preferred over the pauses-at-punctuation only method is likely to be related with the higher number of pauses in linguistically reasonable locations. These pauses helped subjects to better parse the speech stream into manageable chunks, process the information more easily and boost naturalness.

The biggest problem in the pause prediction challenge remains the avoidance of unacceptable pause insertions. A single pause in an unreasonable location can result in a significant downgrading in a preference test. This is likely to be a result of a higher cognitive effort to parse/understand the content of the message.

Regarding the question whether in speech synthesis it is better to over-predict or under-predict pauses, recent research in neuro-psychology suggests that "superfluous prosodic breaks lead to more processing problems than missing ones." [18], i.e. it might be better to slightly under-predict than over-predict, although the current results do show that subjects preferred more pauses than inserted by a pauses-at-punctuation baseline. This might be an indication that there is a trade-off between pause rate and pause locations in speech synthesis.

A typical procedure to train a pause predictor is to use a sentence-level TTS corpus from a single speaker as training material. Such a corpus is usually automatically aligned at the phone-level, which includes the alignment of silences (typically non-speech intervals before the beginning and after the end of the sentence) and pauses (silence or non-speech intervals within the sentence). Often a manual checking is conducted to eliminate any mis-alignments. To train a pause predictor a feature vector is created for each word, including features which can be extracted from text, i.e. the presence or absence of punctuation after a word or whether the word is a content word or a function word. This feature vector is then combined with the annotated pause locations and a machine learning algorithm is used to learn from a subset of this data (typically split into 90% train and 10% test data) to predict pauses for unseen text. However, quite often some of the features used in pause prediction are a result of other predictors, like part-of-speech taggers and parsers which have limited accuracy themselves. This does have a knock-on effect on the pause predictor when using these "noisy" features which potentially can lead to mis-predictions. A widely used feature for pause prediction are part-of-speech tags (PoS). Taggers parse the input sentence and generate a PoS sequence, which is often the most likely sequence given the training material the tagger has been trained on. While the accuracy of modern taggers is in the region above 95% there are still mis-predictions and similar mis-predictions may happen for other features used in pause prediction.

One of the strategies in PauLo to counteract the problems of "noisy" input features is to re-evaluate them in a scoring system and to introduce a mixture of rules to post-filter earlier decisions and finally reconsider any pause sequence in order to enable the algorithm to delete or insert pauses. This way PauLo combines machine learning algorithms with rule-based methods with the goal to improve prediction performance.

5. Conclusion

A new pause prediction method was presented. This method uses a multi-stage approach to predict pauses from text. In the first stage, the position of prosodic chunk boundaries is predicted, then, the second stage re-evaluates these prosodic breaks together with other features in a scoring system and a sequence evaluation step finally predicts pause locations. The new approach also enables users to specify the rate of pause insertions by setting a threshold value and enables a multi-level output as well as the output of a numeric score between 0 - 1 reflecting the strength of a pause.

Objective and subjective evaluations showed that the new model was significantly preferred over a pauses-at-punctuation-only baseline. However, performance did still not achieve the level of systems which had pauses inserted by humans.

Future research needs to look into the accuracy of the input features, especially the prediction of part-of-speech tags to boost the overall accuracy of the pause predictor. Semantic and pragmatic features will have to be included to reach a higher level of performance.

The consideration of pause duration is another subject for future work, especially its link with pause strength levels which in turn are another evolution of the current system.

6. References

[1] C. Sorin, D. Larreur, and R. Llorca, "A rythm-based prosodic parser for text-to-speech systems," in *Proc. of 11th International Congress of Phonetic Sciences, ICPhS, Talinn, Estonia*, 1987, pp. 125–128.

[2] J. Bachenko and E. Fitzpatrick, "A computational grammar of discourse-neutral prosodic phrasing in English," *Computational Linguistics*, vol. 16(3), pp. 155–170, 1990.

[3] M. Atterer, "Assigning prosodic structure for speech synthesis: a rule-based approach," in *Proc. of Speech Prosody 2002, Aix-en-Provence, France, April 11-13*, 2002.

[4] M. Ostendorf and N. Veilleux, "A hierarchical stochastic model for automatic prediction of prosodic boundary location," *Journal of Computational Linguistics, vol. 20, issue 1*, pp. 26–53, 1989.

[5] P. Taylor and A. W. Black, "Assigning phrase breaks from part-of-speech sequences," *Computer Speech and Language*, vol. 12, pp. 99–117, 1998.

[6] P. Bell, T. Burrows, and P. Taylor, "Adaptation of prosodic phrasing models," in *Speech Prosody 2006 – Proceedings of the 3rd Speech Prosody, 2-5 May 2006, Dresden, Germany*, 2006.

[7] V. Keri, S. C. Pammi, and K. Prahallad, "Pause prediction from lexical and syntax information," in *Proceedings of International Conference on Natural Language Processing (ICON)*, 2007.

[8] A. Rosenberg, R. Fernandez, and B. Ramabhadran, "Modeling phrasing and prominence using deep recurrent learning," in *Proceedings of Interspeech 2015 – 16th Annual Conference of the International Speech Communication Association, 6-10 September, Dresden, Germany*, 2015, pp. 3066–3070.

[9] I. Read and S. Cox, "Stochastic and syntactic techniques for predicting phrase breaks," *Computer Speech and Language*, vol. 21, no. 3, pp. 519 – 542, 2007.

[10] C. Brierley, "Prosody resources and symbolic prosodic features for automated phrase break prediction," *PhD thesis, University of Leeds*, 2011.

[11] F. Liu and Y. Zhou, "Tree-guided transformation-based intonational phrase break prediction," in *2011 International Conference in Electrics, Communication and Automatic Control Proceedings*, R. Chen, Ed. Springer New York, 2012, pp. 811–817.

[12] T. T. Nguyen, G. Neubig, H. Shindo, S. Sakti, T. Toda, and S. Nakamura, "A latent variable model for joint pause prediction and dependency parsing," in *Proceedings of Interspeech 2015 – 16th Annual Conference of the International Speech Communication Association, 6-10 September, Dresden, Germany*, 2015, pp. 2719–2723.

[13] Q. Chen, Z. Ling, C. Yang, and L.-R. Dai, "Automatic phrase boundary labeling of speech synthesis database using context-dependent HMMs and n-gram prior distributions," in *Proceedings of Interspeech 2015 – 16th Annual Conference of the International Speech Communication Association, 6-10 September, Dresden, Germany*, 2015, pp. 1581–1585.

[14] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "ToBI: a standard for labelling English prosody," in *Proceedings of International Conference on Spoken Language Processing*, 1992, pp. 12–16.

[15] T. Ingulfsen, "Influence of syntax on prosodic boundary prediction," *Tech. Rep. UCAM-CL-TR-610 University of Cambridge Computer Laboratory*, 2004.

[16] J. Tauberer, "Predicting intrasentential pauses: Is syntactic structure useful?" in *Proceedings of Speech Prosody 2008 – 4th Speech Prosody, 6-9 May 2008, Campinas, Brazil*, 2008.

[17] K. Steinhauer and A. Friederici, "Prosodic boundaries, comma rules, and brain responses: the closure positive shift in ERPs as a universal marker for prosodic phrasing in listeners and readers," *Journal of Psycholinguistic Research*, vol. 30, no. 3, 2001.

[18] S. Bögels, H. Schriefers, W. Vonk, D. J. Chwilla, and R. Kerkhofs, "Processing consequences of superfluous and missing prosodic breaks in auditory sentence comprehension," *Neuropsychologia*, vol. 51, no. 13, pp. 2715 – 2728, 2013.

[19] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence segmentation and punctuation recovery for spoken language translation," in *Proceedings of the International Conference of Acoustics, Speech and Signal Processing, ICASSP, 31 March - 4 April, Las Vegas, USA*, 2008, pp. 5105–5108.

[20] J. Miranda, J. Neto, and A. Black, "Improved punctuation recovery through combination of multiple speech streams," *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 132 – 137, 2013.

[21] A. Parlikar and A. W. Black, "Minimum error rate training for phrasing in speech synthesis," in *Proceedings of the 8th ISCA Speech Synthesis Workshop, SSW8, Barelona, Spain*, 2013.

[22] T. Mishra, Y. Kim, and S. Bangalore, "Intonational phrase break prediction for text-to-speech synthesis using dependency relations," in *Proceedings of ICASSP 2015 – 40th International Conference on Acoustics, Speech and Signal Processing, 19-24 April, Brisbane, Australia*, 2015, pp. 4919–4923.

[23] J. R. Quinlan, *C4.5: Programming for Machine Learning*. San Mateo, CA: Morgan Kaufman, 1993.

[24] T. Burrows, P. Jackson, K. Knill, and D. Sityaev, "Combining models of prosodic phrasing and pausing," in *Proceedings Interspeech 2005 – 9th Annual Conference of the International Speech Communication Association, 4-8 September, Lisboa, Portugal*, 2005, pp. 1829–1832.

[25] HARVARD sentences. [Online]. Available: http://www.cs.columbia.edu/˜hgs/audio/harvard.html