CROWDSOURCED AND AUTOMATIC SPEECH PROMINENCE ESTIMATION

Max Morrison, Pranav Pawar, Nathan Pruyne, Jennifer Cole, Bryan Pardo

Northwestern University, Evanston, IL, USA

ABSTRACT

The prominence of a spoken word is the degree to which an average native listener perceives the word as salient or emphasized relative to its context. Speech prominence estimation is the process of assigning a numeric value to the prominence of each word in an utterance. These prominence labels are useful for linguistic analysis, as well as training automated systems to perform emphasis-controlled text-to-speech or emotion recognition. Manually annotating prominence is time-consuming and expensive, which motivates the development of automated methods for speech prominence estimation. However, developing such an automated system using machine-learning methods requires human-annotated training data. Using our system for acquiring such human annotations, we collect and open-source crowdsourced annotations of a portion of the LibriTTS dataset. We use these annotations as ground truth to train a neural speech prominence estimator that generalizes to unseen speakers, datasets, and speaking styles. We investigate design decisions for neural prominence estimation as well as how neural prominence estimation improves as a function of two key factors of annotation cost: dataset size and the number of annotations per utterance.

Index Terms: emphasis, paralinguistics, prominence, prosody

1. INTRODUCTION

Prominent or *emphasized* words are those that stand out to listeners as salient or perceptually highlighted relative to their context. Human perception of prominence in English (and many other languages) is influenced by acoustic factors related to *prosody* (i.e., the pitch, rhythm, and loudness of speech) as well as *information structure* [1, 2]. Information structure is the contribution a word makes to the shared knowledge of the speaker and hearer, based on its status as conveying information that is given, new, or contrastive relative to prior discourse. Prominence is a multi-dimensional perceptual relation between words in a phrase. However, performing annotations in this multi-dimensional space (e.g., separately annotating structural and semantic factors), as opposed to a scalar, is cost-prohibitive.

In this paper, we represent the emphasis status of each word using a binary label (zero or one), and its prominence as a scalar real value between zero and one, such that the emphasis status is a binary thresholding of scalar prominence, and scalar prominence is a weighted norm of the latent multi-dimensional prominence. From a statistical viewpoint, we consider the prominence m of word i as a Bernoulli distribution parameter and the emphasis e as the corresponding Bernoulli random variable such that the probability that word i is perceived as emphasized is $p(e_i = 1) = m_i$.

Emphasis annotation is the process of labeling each word with a binary indicator of emphasis, a task that can be performed by non-expert native speakers. Given multiple annotations of the same speech (e.g., from multiple human annotators), scalar prominence values can be obtained by averaging over the binary emphasis annotations for each word. Emphasis and prominence labels are used in downstream tasks such as emphasis-controlled TTS [3, 4, 5], emotion recognition [6], and text summarization [7].

Because human emphasis annotation can be costly and time-consuming, prior works attempted to replace these annotations with one of three types of automated methods: (1) heuristic, rule-based methods based on acoustic features [8, 9, 10], (2) machine learning methods that train on the output of such a rule-based system [11, 12, 13], and (3) machine learning methods that train on ground-truth values prominence derived from human annotation [14, 15, 16, 17]. Heuristic, rule-based systems struggle to capture the complexity of high-dimensional perceptual attributes and can require significant manual tuning to generalize to new data distributions. Machine learning methods that learn from the output of rule-based systems might perform useful interpolation and denoising, but otherwise inherit the same drawbacks. Methods that learn from human annotation avoid these drawbacks, but require human-annotated data for training. All three approaches necessitate benchmarking on human annotations, making human annotation unavoidable.

Vaidya et al. [17] is most similar to our method of performing neural prominence estimation using human annotations. They use a closed-source dataset of 41k English words spoken by 10-14 year old students in Mumbai and annotated by 26 university students to have seven annotations per word. Transcripts come from 34 short stories of 100 words, each selected to have a single reference prosody for fluency assessment purposes [18]. In other words, text was selected to make placement of emphasis in the dataset more predictable. Vaidya et al. train a CRNN-based model to infer prominence from ground truth acoustic features with a Pearson correlation of 0.721 on heldout recordings of unseen non-native child speakers. They demonstrate further performance using human-annotated boundary features, language-dependent lexical features, and curated selection of prosodic features. Lack of open source data, models, and input features makes exact replication impossible. Instead, we perform ablations of what we consider to be key architectural decisions and note the additional utility of the input features they have explored.

Unique to our work, we train on scalable crowdsourced annotations, generalize to unseen speaking styles and datasets of adult speakers, produce open-source annotations suitable for training and benchmarking, develop open-source tools for crowdsourced speech annotation, and uncover novel guidelines for how the performance of neural prominence estimation trained on crowdsourced annotations scales with dataset size and annotator redundancy—two primary factors in the overall cost of emphasis annotation. Our main contributions in this paper are as follows.

- **(Contribution 1)** We develop a neural speech prominence estimation system trained on crowdsourced human emphasis annotations that produces accurate prominence estimations on unseen speakers, datasets, and speaking styles (Section 4).

- **(Contribution 2)** We produce a CC-BY-4.0 licensed dataset of emphasis annotations of one eighth of the

Instructions

Listen to the audio file a minimum of two times. Select the words that were emphasized by the speaker. The emphasized words are those that stand out from nearby words. Play the audio and then click on a word to select (boldface) or deselect it.

▶ 0:05 / 0:05 ━━━━━━ 🔊 ⋮

AS   TO   **THAT**   MR   **CHASE**   I   **WILL**   TELL   MY   FATHER   **ALL**   ABOUT

IT   AS   **SOON**   AS   HE   COMES   HOME   FROM   THE   **BANK**

Next

Fig. 1. **Crowdsourced human emphasis annotation interface** | Annotators listen to a speech recording and click to boldface the words they perceive as emphasized.

`train-clean-100` partition of LibriTTS (Section 3).

- **(Contribution 3)** We develop an open-source system for performing crowdsourced word-level annotations of, e.g., falsetto, vocal fry, and emphasis (Section 2).

- **(Contribution 4)** We demonstrate how the amount of training data and the number of annotators per speech excerpt impact estimation performance, providing guidelines for cost-effective annotation (Section 6).

We release our code and annotation methods as `emphases`, a MIT-licensed, pip-installable Python module for training, evaluating, and performing both automatic and human annotation of emphasis. Our code and dataset are available on our project website.[1]

2. CROWDSOURCING EMPHASIS ANNOTATION

Here, we describe our open-source crowdsourced annotation tool for annotation of, e.g., prominence, mispronunciation, or vocal fry annotation. We developed our human annotation system as a word selection task that we add to Reproducible Subjective Evaluation (ReSEval) [19]. ReSEval is a subjective evaluation tool that handles database, server, and crowdsourced participant acquisition to quickly create and manage crowdsourced evaluations in Python. ReSEval enables a greater variety of tasks and prescreening criteria (e.g., listening tests) than existing survey templates, such as those available on Amazon Mechanical Turk (MTurk). For prominence annotation, we first require annotators to pass a listening test that ensures a suitable listening environment, using the listening test method proposed by Cartwright et al. [20]. We then present annotators with an audio recording and the corresponding text (see Figure 1). As in the annotation interface of Cole et al. [21], annotators are required to listen to the audio file a minimum of two times and asked to select all of the words that were emphasized by the speaker by clicking on the words themselves. Annotators must start the audio to begin selecting words, but may select words while the audio is playing.

---

[1]`maxrmorrison.com/sites/prominence-estimation/`

3. EMPHASIS ANNOTATION DATASET

We used our crowdsourced annotation system to perform human annotation on one eighth of the `train-clean-100` partition of the LibriTTS [22] dataset. Specifically, participants annotated 3,626 utterances with a total length of 6.42 hours and 69,809 words from 18 speakers (9 male and 9 female). We collected at least one annotation of all 3,626 utterances, at least two annotations of 2,259 of those utterances, at least four annotations of 974 utterances, and at least eight annotations of 453 utterances. We did this in order to explore (in Section 6) whether it is more cost-effective to train a system on multiple annotations of fewer utterances or fewer annotations of more utterances. We paid 298 annotators to annotate batches of 20 utterances, where each batch takes approximately 15 minutes. We paid $3.34 for each completed batch (estimated $13.35 per hour). Annotators each annotated between one and six batches. We recruited on MTurk US residents with an approval rating of at least 99 and at least 1000 approved tasks. Today, microlabor platforms like MTurk are plagued by automated task-completion software agents (bots) that randomly fill out surveys. We filtered out bots by excluding annotations from an additional 107 annotators that marked more than $2/3$ of words as emphasized in eight or more utterances of the 20 utterances in a batch. Annotators who fail the bot filter are blocked from performing further annotation. We also recorded participants' native country and language, but note these may be unreliable as many MTurk workers use VPNs to subvert IP region filters on MTurk [23].

The average Cohen Kappa score for annotators with at least one overlapping utterance is 0.226 (i.e., "Fair" agreement)—but not all annotators annotate the same utterances, and this overemphasizes pairs of annotators with low overlap. Therefore, we use a one-parameter logistic model (i.e., a Rasch model) computed via `py-irt` [24], which predicts heldout annotations from scores of overlapping annotators with 77.7% accuracy (50% is random).

4. NEURAL PROMINENCE ESTIMATION

We propose a neural network that predicts human-annotated prominence values for a sequence of words from acoustic features. Our proposed model takes as input an 80-channel Mel spectrogram) at
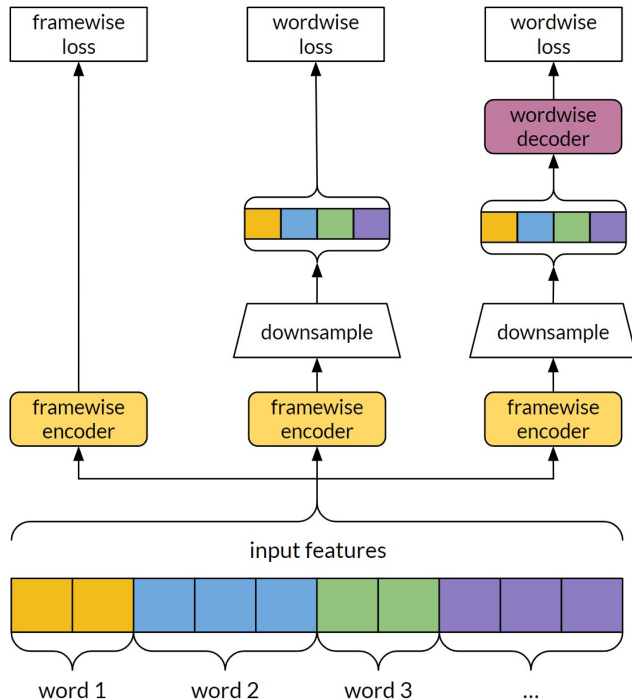
Fig. 2. **Three candidate prominence estimation models** | We experiment with a **framewise** model (**left**) as well as two wordwise models: one that downsamples from frames to words just before the loss function (**posthoc wordwise**; **center**) and one that downsamples within the neural network (**intermediate wordwise**; **right**). The yellow, framewise encoder is a stack of convolution layers that operate at the frame resolution. The magenta, wordwise decoder is a stack of convolution layers that operate at the word resolution.

an evenly-quantized time frame resolution (e.g., ten milliseconds), as well as a time-alignment between words and frames (i.e., the start and end frame indices corresponding to each word). We found no improvement when adding pitch, periodicity, or loudness features, which may be redundant with the spectrogram. Our network produces one prominence value per word. Crucial to such a system is the mechanism to perform variable-stride downsampling from the frame resolution to the word resolution. Vaidya et al. [17] propose one option (which we refer to as **prehoc wordwise**) that segments words before input to the model, such that the receptive field of the framewise encoder is limited to a single word. We identify three additional locations within the system for performing downsampling (Figure 2): (1) (**framewise**) upsample the ground truth prominence values to the frame resolution using linear interpolation during training and downsample the network output to word resolution during inference, (2) (**posthoc wordwise**) downsample the network output during training and inference, and (3) (**intermediate wordwise**) downsample to the word resolution within the network. We further identify four methods for variable-stride downsampling from the frame resolution to the word resolution at each location: (1) (**average**) take the channel-wise average over all frames corresponding to a word, (2) (**max**) take the channel-wise maximum over all frames corresponding to a word, (3) (**sum**) take the channel-wise sum over all frames corresponding to a word, and (4) (**center**) take the value of the frame in the center of the word. Each framewise encoder and wordwise decoder is a stack of six convolution

Downsampling location	Downsampling Method			
	Average	Center	Max	**Sum**
Inference (framewise)	0.102	0.153	0.102	0.137
Intermediate (wordwise)	0.656	0.438	0.674	**0.675**
Posthoc (wordwise)	0.440	0.385	0.623	0.645
Prehoc [17] (wordwise)	0.670	0.471	0.670	0.656

Table 1. **Ablations of downsampling methods and locations** | Pearson correlations between estimated and ground truth prominence on the unseen Buckeye [29, 21] dataset. Averages over three runs.

layers with 80 channels, a kernel size of three, and intermediate ReLU activation. Our hyperparameter search indicates six layers improves over five or seven layers; 80 channels improves over 64 or 128 channels; ReLU improves over leaky ReLU [25], GeLU [26], or Swish [27]; convolution layers improve over Transformer [28] layers; and dropout does not improve performance when our model is not overparameterized.

5. EVALUATION

We design our evaluation to determine whether our proposed neural prominence estimation models exceed the performance of previous automatic prominence estimation methods. We perform ablations to show the relative impact of design decisions, such as the method for downsampling from frames to words. Finally, we demonstrate the scaling behaviors of our best model as a function of two key cost factors: the number of emphasis-annotated utterances and the number of annotators per utterance.

5.1. Data

We train our models on our annotated LibriTTS partition (Section 2). We determine word boundaries by performing forced alignment between the speech transcript and audio. We use the Penn Phonetic Forced Aligner (P2FA) [30] via the Python Forced Alignment (`pyfoal`) library [31]. We extract from the 16 kHz audio 80 bands of a log-mel spectrogram using a hopsize of 160 samples and a window size of 1024 samples. We partition utterances into train (80%), validation (10%), and test (10%) partitions. Scaling experiments use a different partitioning (see Section 5.4).

To examine generalization to unseen speakers, datasets, and speaking styles, we perform additional evaluation on emphasis annotations [21] of the Buckeye corpus of conversational American English [29]. This evaluation dataset consists of 16 utterances from 16 speakers for a total of 256 utterances lasting 3.98 minutes and containing 931 words. Each utterance is annotated by 32 native speakers of American English.

5.2. Training

We train using an Adam optimizer with a learning rate of $1e^{-3}$ to optimize a binary cross entropy (BCE) loss between predicted and ground truth prominence values. In Table 2, we show that mean squared error (MSE) and bounding the output to be between zero and one performs comparably with BCE, making MSE a viable option for loss function as well. We train for 6,000 steps, validating every 100 steps and checkpointing when the Pearson correlation between model output and human labeling (Section 5.3) is the maximum so far. We use a variable batch size [32] during training with a maximum of 75,000 frames (12.5 minutes) per batch.

Model	Buckeye		LibriTTS	
	PC↑	BCE↓	PC↑	BCE↓
Proposed	**0.675**	**0.337**	**0.534**	**0.358**
MSE loss	0.672	0.340	0.533	0.361
Wavelet [9, 3]	0.529	–	0.393	–

Table 2. **Automatic prominence estimation results** | Pearson correlation (PC) and binary cross-entropy (BCE) between inferred and human annotations on two datasets. Includes our best model (the **intermediate wordwise** model with **sum** downsampling), one ablation, and a heuristic baseline. Averages over three runs.

5.3. Metrics

We measure the accuracy of a prominence estimation system by calculating the Pearson correlation and BCE between the system's output and the corresponding ground truth prominence value for each word. BCE is equal to the KL divergence between Bernoulli random variables up to a dataset-dependent constant, making it well-suited for our probabilistic view of prominence as Bernoulli distribution parameters (Section 1). We use Pearson correlation (instead of BCE) to compare with digital signal processing (DSP) baselines, as the range of system outputs varies between baseline systems and Pearson correlation is scale-invariant.

5.4. Experimental design

We now describe our experiments designed to demonstrate the performance of our model relative to previous works, the efficacy of our individual design choices, and the scaling behaviors of interest.

In Table 1, we experiment with all combinations of the four downsampling locations and four downsampling methods described in Section 4. In Table 2, we ablate our loss function and compare to a top-performing heuristic method (**wavelet**) [3] that classifies words by thresholding wavelet-based features proposed by Suni et. al [9].

We demonstrate scaling behaviors for prominence estimation using our best model on curated data partitions of 400, 800, 1,600, and 3,200 utterances. We further analyze the cost efficiency of annotator redundancy by training on 400 annotations with eight annotators, 800 annotations with four annotators, 1,600 annotations with two annotators, and 3,200 annotations with one annotator. We discuss implications for cost-effective annotation.

6. RESULTS

The accuracies of our proposed neural prominence estimation method as well as baseline and ablation systems are reported in Tables 1 and 2. In Table 1, we see that the variable-stride downsampling required for converting frame-resolution acoustic features to word-resolution prominence estimations is best performed within the network (**intermediate**) by taking the **sum** of each channel over the frames corresponding to each word. This outperforms the method proposed by Vaidya et al. [17] (**prehoc**) that segments the input acoustic features into words and indicates that it is efficacious for the receptive field of the framewise encoder to span across words. Relative to **prehoc**, our proposed **intermediate** method is also faster to train (by 39.7%) and reduces GPU memory consumption (by 69.1%) during training. In Table 2, we see that our best model significantly outperforms a top heuristic baseline (**wavelet**) in PC and BCE on both heldout data from our LibriTTS annotations and heldout data from an unseen dataset with speakers and speaking styles outside the training distribution (Buckeye).
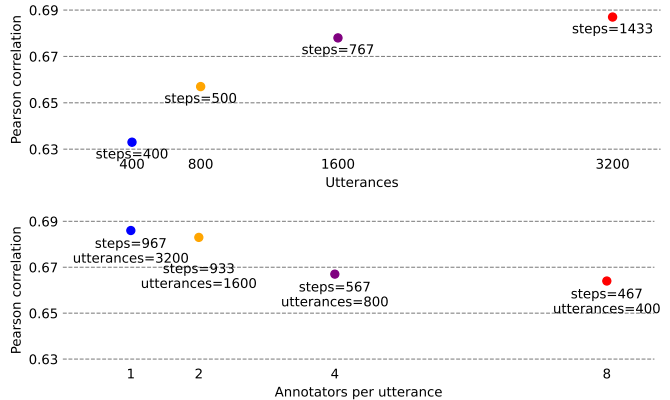
Fig. 3. **Scaling behaviors for neural prominence estimation** | **(top)** Pearson correlation between inferred and human annotations as a function of dataset size. **(bottom)** Pearson correlation as a function of number of annotators for a fixed annotation budget. Averages over three runs.

Figure 3 demonstrates scaling behaviors for neural prominence estimation using our crowdsourced emphasis annotations of the LibriTTS dataset. We provide key takeaways:

For a fixed budget, use only one annotator per utterance | Neural networks benefit from training on larger datasets. This benefit empirically outweighs the benefit of variance reduction in the ground truth distribution caused by annotator redundancy. However, when data is limited, annotator redundancy can improve performance. Cole et al. [21] note marginal variance reduction beyond seven annotators, so we do not expect these improvements to extrapolate to, e.g., 16 annotations per utterance.

For a fixed budget, increasing annotations per utterance up to eight improves convergence speed | As the number of annotations per utterance increases from one to eight, the number of steps needed for convergence significantly decreases.

To further examine scaling behaviors, we used our best model to annotate the entire `train-clean-100` partition of LibriTTS and train a new model from scratch on 26,588 automatically annotated utterances. The resulting model performed marginally worse on heldout data, underscoring the utility of human annotations.

7. CONCLUSION

The prominence of a spoken word is a fundamentally perceptual phenomenon. Our work highlights the benefits of utilizing human perception for prominence estimation, and demonstrates high-quality, generalizable prominence estimation trained from crowdsourced emphasis annotations (**Contribution 1**). We further solve the lack of publicly available prominence annotations suitable for training a generalizable machine learning model (**Contribution 2**) and provide tools (**Contribution 3**) and guidelines (**Contribution 4**) for practitioners performing crowdsourced annotation. These contributions enable future work in high-quality emphasis-controlled text-to-speech, analysis of the human perception of prominence, and automatic detection and control of other word-level attributes (e.g., disfluency, falsetto, and vocal fry).

8. REFERENCES

[1] Jennifer Cole, José I. Hualde, Caroline L. Smith, Christopher Eager, Timothy Mahrt, and Ricardo Napoleão de Souza, "Sound, structure and meaning: The bases of prominence ratings in English, French and Spanish," *Journal of Phonetics*, 2019.

[2] Yong cheol Lee, Bei Wang, Sisi Chen, Martine Adda-Decker, Angélique Amelot, Satoshi Nambu, and Mark Liberman, "A crosslinguistic study of prosodic focus," in *International Conference on Acoustics, Speech, and Signal Processing*, 2015.

[3] Shreyas Seshadri, Tuomo Raitio, Dan Castellani, and Jiangchuan Li, "Emphasis control for parallel neural TTS," in *Interspeech*, 2022.

[4] Antti Suni, Sofoklis Kakouros, Martti Vainio, and Juraj Šimko, "Prosodic prominence and boundaries in sequence-to-sequence speech synthesis," in *Speech Prosody*, May 2020.

[5] Sophie Roekhaut, Jean-Philippe Goldman, and Anne Catherine Simon, "A model for varying speaking style in TTS systems," in *Speech Prosody*, 2010.

[6] K. Sreenivasa Rao, Shashidhar G. Koolagudi, and Ramu Reddy Vempada, "Emotion recognition from speech using global and local prosodic features," *International Journal of Speech Technology*, 2013.

[7] Yanju Chen and Rong Pan, "Automatic emphatic information extraction from aligned acoustic data and its application on sentence compression," *AAAI Conference on Artificial Intelligence*, 2017.

[8] Sofoklis Kakouros and Okko Räsänen, "3PRO – An unsupervised method for the automatic detection of sentence prominence in speech," *Speech Communication*, 2016.

[9] Antti Suni, Juraj Šimko, Daniel Aalto, and Martti Vainio, "Hierarchical representation and estimation of prosody using continuous wavelet transform," *Computer Speech & Language*, 2017.

[10] Slava Shechtman, Raul Fernandez, and David Haws, "Supervised and unsupervised approaches for controlling narrow lexical focus in sequence-to-sequence speech synthesis," in *IEEE Spoken Language Technology Workshop*, 2021.

[11] Zofia Malisz, Harald Berthelsen, Jonas Beskow, and Joakim Gustafson, "Controlling prominence realisation in parametric DNN-based speech synthesis," in *Interspeech*, 2017.

[12] Aarne Talman, Antti Suni, Hande Celikkanat, Sofoklis Kakouros, Jörg Tiedemann, and Martti Vainio, "Predicting prosodic prominence from text with pre-trained contextualized word representations," in *Nordic Conference on Computational Linguistics*, 2019.

[13] Brooke Stephenson, Laurent Besacier, Laurent Girin, and Thomas Hueber, "BERT, can HE predict contrastive focus? predicting and controlling prominence in neural TTS using a language model," in *Interspeech*, 2022.

[14] Taniya Mishra, Vivek Rangarajan Sridhar, and Alistair Conkie, "Word prominence detection using robust yet simple prosodic features," in *Interspeech*, 2012.

[15] George Christodoulides, Mathieu Avanzi, and Anne Catherine Simon, "Automatic labelling of prosodic prominence, phrasing and disfluencies in French speech by simulating the perception of naïve and expert listeners," in *Interspeech*, 2017.

[16] Sabrina Stehwien, Antje Schweitzer, and Ngoc Thang Vu, "Acoustic and temporal representations in convolutional neural network models of prosodic events," *Speech Communication*, 2020.

[17] Mithilesh Vaidya, Kamini Sabu, and Preeti Rao, "Deep learning for prominence detection in children's read speech," in *International Conference on Acoustics, Speech and Signal Processing*, 2022.

[18] Kamini Sabu and Preeti Rao, "Prosodic event detection in children's read speech," *Computer Speech & Language*, 2021.

[19] Max Morrison, Brian Tang, Gefei Tan, and Bryan Pardo, "Reproducible subjective evaluation," in *ICLR Workshop on ML Evaluation Standards*, 2022.

[20] Mark Cartwright, Bryan Pardo, Gautham J Mysore, and Matt Hoffman, "Fast and easy crowdsourced perceptual audio evaluation," in *International Conference on Acoustics, Speech and Signal Processing*, 2016.

[21] Jennifer Cole, Timothy Mahrt, and Joseph Roy, "Crowdsourcing prosodic annotation," *Computer Speech & Language*, 2017.

[22] Heiga Zen, Rob Clark, Ron J. Weiss, Viet Dang, Ye Jia, Yonghui Wu, Yu Zhang, and Zhifeng Chen, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in *Interspeech*, 2019.

[23] Aaron J Moss, Cheskie Rosenzweig, Shalom N Jaffe, Richa Gautam, Jonathan Robinson, and Leib Litman, "Bots or inattentive humans? Identifying sources of low-quality data in online platforms," *PsyArXiv preprint PsyArXiv:wr8ds*, 2021.

[24] John Patrick Lalor and Pedro Rodriguez, "py-irt: A scalable item response theory library for Python," *INFORMS Journal on Computing*, 2023.

[25] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al., "Rectifier nonlinearities improve neural network acoustic models," in *International Conference on Machine Learning*, 2013.

[26] Dan Hendrycks and Kevin Gimpel, "Gaussian error linear units," *arXiv preprint arXiv:1606.08415*, 2016.

[27] Stefan Elfwing, Eiji Uchibe, and Kenji Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural networks*, 2018.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017.

[29] Mark A Pitt, Keith Johnson, Elizabeth Hume, Scott Kiesling, and William Raymond, "The Buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability," *Speech Communication*, 2005.

[30] Jiahong Yuan and Mark Liberman, "Speaker identification on the SCOTUS corpus," *The Journal of the Acoustical Society of America*, 2008.

[31] Max Morrison, "Python forced alignment (version 0.0.3)," https://github.com/maxrmorrison/pyfoal, 2023.

[32] Philippe Gonzalez, Tommy Sonne Alstrøm, and Tobias May, "On batching variable size inputs for training end-to-end speech enhancement systems," *arXiv preprint arXiv:2301.10587*, 2023.