ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10448289

VOICE TOXICITY DETECTION USING MULTI-TASK LEARNING

Mahesh Kumar Nandwana[*,1], *Yifan He*[*†,1,2], *Joseph Liu*[1], *Xiao Yu*[1],
Charles Shang[1], *Eloi Du Bois*[1], *Morgan McGuire*[1], *Kiran Bhat*[1]

[1]Roblox, San Mateo, CA [2]Carnegie Mellon University, Pittsburgh, PA

ABSTRACT

Social communication systems must identify toxic voice audio to support moderation that protects the safety and civility of their communities. Toxicity classification for voice depends on both audio style, such as volume and tone, and content, such as the words in the speech individually and in context. We introduce a novel end-to-end multi-task learning (MTL) paradigm for audio-based toxicity detection, addressing the challenges associated with existing automatic speech recognition (ASR) and text-based systems. By employing a hard parameter-sharing backbone and flexible soft-attention task adapters, our model performs two tasks: a multi-label toxicity classification task that targets specific categories of toxic behavior, and an auxiliary Audio to Keyword detection task that focuses on transcribing only toxic keywords, thereby enhancing computational efficiency and complementing classification output. We observe that the classifier significantly improves the quality of keyword detection. We also contribute a data pipeline for automated offline labeling of training sets.

Index Terms— Toxicity Detection, Multi-task Learning, Speech Recognition, Audio to Keyword Detection.

1. INTRODUCTION

Voice chat on platforms such as online gaming and live chat rooms has become an integral communication medium, facilitating real-time interactions. This presents a significant challenge in moderating potentially toxic audio content to maintain safety and civility on online platforms [1]. Most of the previous systems are text-based classifiers [2, 3, 4, 5], adapted for speech by using an Automatic Speech Recognition (ASR) component to transcribe the audio into text where these classifiers are then run [6]. Such methods are effective, but incur huge computational costs due to the expensive computation requirements of ASR, and require different components of the system to be independently trained, making it difficult for widespread use. Moreover, the real-time nature of toxic speech means that it is difficult for such solutions to scale to millions of users.

In response to these challenges, some recent audio-based methods attempt to classify toxicity directly on audio [7, 8, 9], reporting promising classification results on small-scale datasets. However, such approaches are hamstrung by the lack of large scale training data. Public datasets such as DeToxy [6] and IEMOCAP [10] have extremely small balanced datasets and lack real-world characteristics. Even in cases such as Yousefi and Emmanouilidou [7], where there is access to larger scale internal training datasets up to hundreds of hours of data, models are trained and evaluated on binary labels (toxic/non-toxic), which does not highlight how the model performs across different types of toxic content.

Different approaches have been used to improve speech-based toxicity classification methods, such as custom attention architectures to model semantic information [7], or using pre-trained speech encoders such as Wav2Vec2.0 [11] to improve robustness. Multi-task learning (MTL) has been also proposed as an approach for improving the performance of the model on these toxicity classifiers, such as using ASR as an auxiliary task to condition the encoder while it is jointly trained for toxicity classification [9], but the task itself is not used towards toxicity detection.

This paper proposes an end-to-end multi-task model that attempts to use multi-task learning with both tasks directed at toxicity detection. The first task is a multi-label task that classifies 5 different genres of toxicity, while the second task is a keyword detection task that is formulated as a limited-vocabulary ASR task, so only keywords relevant to toxicity are only transcribed explicitly. We also propose alternative MTL architectures that also attempt to leverage the dynamics between different toxicity classes and keywords. With these two tasks, this paper is able to contribute the following. Firstly, we introduce a novel multi-task learning paradigm for audio-based toxicity detection, where both tasks help contribute towards moderation, as keyword detection can be used to explain certain toxicity classification decisions. Secondly, we present empirical experimental results on large-scale real-world datasets compared to previous works on much smaller datasets, underscoring the robustness and practical applicability of the proposed model in industry deployment. Thirdly, our multi-label problem formulation for toxicity classification highlights how not all types of toxicity are equal, and how dataset scale and curation can influence model evaluation metrics. Fourthly, through the different MTL variants we propose, we highlight how biasing the toxicity classifier with keyword detection features via different methods of parameter sharing influence the classifier's performance on different types of toxicity. We also show, with minor penalties to specific classes, that we can get better at toxic keyword detection with multi-task learning.

2. DATA PIPELINE

Human data labeling is slow and expensive. For toxic communication, it is also desirable to minimize the exposure of toxic content to human labellers. We developed a scalable automated data pipeline that produces audio labeled with toxicity classes and keywords. We applied this to audio gathered from the Roblox Voice Chat product to train our model.

The data pipeline comprises three stages shown in Figure 1: chunk splitting determined by the Opus codec's DTX extension;

[*]Equal contribution
[†]Work done at Roblox

The authors would like to thank Quoc Le, Jason Golubock, Will Welch, and Hao-En Sung for their help with the Data Pipeline.

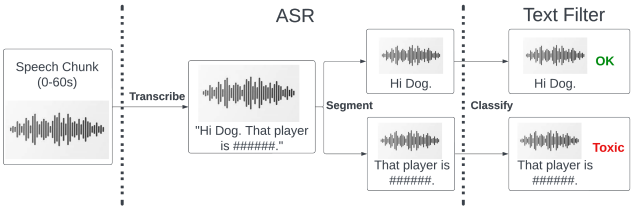

Fig. 1. Data Pipeline for Automatic Annotation of Speech Utterances.

ASR implemented with the publicly available models [12]; and text classification with the Roblox Text Filter.

We pre-process the raw audio into chunks of continuous speech. These are transcribed and segmented into audio phrases using the ASR models. It then applies Roblox's Text Filter for classification. The Text Filter is an ensemble model trained on human-labeled toxic text data comprising an extended DistilBERT model [13, 14] and regular expressions. Finally, chunks that the Text Filter classifies as toxic are annotated with keywords from the ASR transcription.

3. PROPOSED SYSTEM

In this section, we describe the toxicity systems developed for our experiments. We propose three different systems, which include an audio-only toxicity classifier, an audio to keyword system, and a multi-task learning system. We used WavLM [15] as the encoder for our three systems. WavLM is a transformer-based model that learns a universal speech representation from massive unlabeled data in a self-supervised manner. The model is composed of multiple layers of CNNs for local feature extraction and transformer-based blocks for global context modeling. We chose WavLM since it presented the state-of-the-art performance on SUPERB benchmark for almost all the tasks [16].

3.1. Audio to Toxicity Classifier

The Toxicity Classifier is a multi-label model with six categories: Profanity, Bullying, Dating & Sexting, Racism, Other, and No-Violation. The Other category consists of a wide variety of toxic speech such as Grooming, Drugs and Alcohol references, Radicalization, etc. that do not fall cleanly into the first four toxic categories which the text toxicity classifier picks up. Given that a single audio clip can embody multiple types of violations, the task inherently becomes multi-label rather than a conventional multi-class classification problem. We fine-tuned the entire network including head layers for this task with Cross-Entropy (CE) loss.

3.2. Audio to Keyword Detection

To improve the explainability of toxicity classification, we designed a keyword detector that directly operates on the audio signal and localizes specific keywords. This task diverges from conventional ASR systems by focusing on a predefined list of keywords that are congruent with the toxic categories outlined in the Toxicity Classifier task. The ground truth transcriptions are replaced by these keywords in a word-by-word fashion. Words not in the keyword list are substituted by a dummy word, specifically "good" in our experiments.

The task serves multiple objectives. 1) *Focused Information Extraction*: Contrary to conventional ASR systems that transcribe audio signals to every possible word, Audio to Keyword filters out noise by only concentrating on a subset of words deemed relevant for toxicity classification. This not only facilitates better generalization but also diminishes hypothesis space. 2) *Task Synergy*: We hypothesize that the close semantic relationship between the Toxicity Classifier and Audio to Keyword tasks aids in parameter sharing during the multi-task learning process can yield positive inductive bias for both tasks. 3) *Scope Adaptability*: This predefined keyword list also allows for incremental keyword addition or removal, allowing for dynamic performance tuning of the toxicity classifier. Since the goal of audio to keyword is to detect toxic words, general word error rate or character error rate are not pursued, instead we use weighted AP and F1 score of detected words as evaluation metrics.

Compared to other relevant methods like keyword spotting [17, 18] that identifies the presence of keywords, our sequence-to-sequence task is trained on datasets that contain a larger audio context with word by word ordering, which allows our system to predict the location of toxic words. We found that using a simple CTC loss to directly map the acoustic frames to character sequences worked well for our experiments and avoids some of the complexity associated contextual biasing approaches [19]. Notably, to account for the spelling errors commonly associated with CTC decoding, an N-gram language model [20] is employed during the decoding process, thereby minimizing transcription errors.

3.3. Multi-task Learning

3.3.1. Model Architecture

The architecture of the proposed multi-task learning model (MTL) is illustrated in Figure 2(a). We use pre-trained WavLM Base+ fine-tuned on 100 hours of LibriSpeech clean as the shared base backbone. The output logits from the encoder are partitioned into two sets of logits for each individual task. These are then directed towards two distinct task-specific heads, one for multi-label toxicity classification and another for audio-to-keyword detection. This is the basic configuration for MTL where the two tasks share the encoder backbone and are totally decoupled during the decoding process, and serves as a benchmark for other MTL variants described later. The advantage of this approach is the limited parameter size increase (less than 0.1M compared with the single classification task) and simpler/faster training.

3.3.2. Task Adapter

A more complex variant of MTL architecture share the features from a common trunk and adapts it to the classification and keyword detection tasks. The task adapter is a soft-attention mask, designed to operate after any encoder block. The features in the shared backbone and the task adapters for each task can be learned jointly to learn the generalization of the shared features across multiple tasks, and simultaneously maximize the task-specific performance. We apply the adapter to each task in order to learn a combination of task-shared global features and task-specific features in a self-supervised manner.

As shown in Figure 2(b), the extracted feature from the CNN layers is first directed into a position embedding layer. We introduce the task adapter combining a convolution layer and a soft-attention layer. The function of this module resembles that of adapter [21, 22], though its application differs. While adapters typically act as lightweight networks within the backbone to learn domain-specific knowledge, our task adapter tunes the features to be task-specific without losing the generality from the shared encoder.

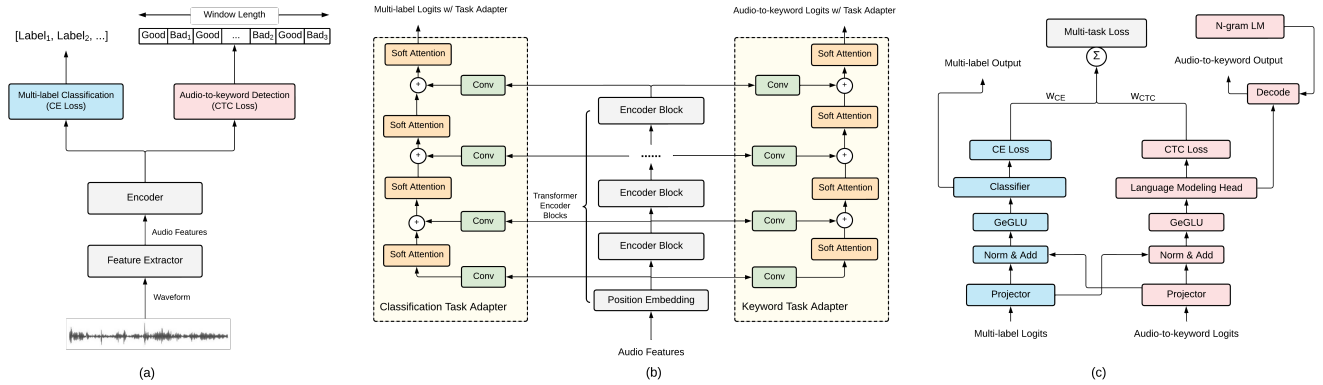

Fig. 2. Proposed MTL architectures: (a) Architecture Overview: feature extractor and encoder in grey is the shared backbone, multi-label toxicity classifier and audio to keyword detection branch are marked in blue and red respectively; (b) Encoder architecture with Task Adapter; (c) Decoder architecture with Information Sharing.

3.3.3. Decoder

Figure 2(c) shows a third MTL variant that operates on the decoder for each task, with the multi-label classification branch optimized using CE loss and the audio to keyword detection branch utilizing CTC loss. The total multi-task loss is computed as their weighted sum. The network's parameters are updated through back-propagation. The outputs from each task's projector are normalized and added together, subsequently serving as the input to their respective final layers prior to undergoing GeGLU activation. We experimented with several combinations of addition and concatenation alongside activation functions including GELU, GeGLU, and SwiGLU. Our empirical results indicate that the combination of addition with GeGLU activation works best in practice. For the basic MTL version mentioned in 3.3.1, the outputs from each task's projector are directly sent to their final layers.

Rather than using task balancing methods such as weight uncertainty [23] or GradNorm [24], we choose to set the weight manually, since we want to have control over the importance of each task. We set different weights for multi-label classification and audio-to-keyword detection as follows:

$$\mathcal{L}_{MTL} = \lambda * \mathcal{L}_{CE} + (1 - \lambda) * \mathcal{L}_{CTC} \quad (1)$$

In the experimental results, we set all λ to 0.7. A N-gram language model is incorporated as an add-on for audio to keyword branch to regularize the decoding process for the language model head, in order to minimize the misspelling errors. In our experiments, we use a 4-gram language model trained from the training transcription corpus.

Profanity	Dating & Sexting	Racist	Bullying	Other	No Violation	Total
Training Dataset (hours)						
1755.1	307.2	225.0	902.5	1,559.5	1791.4	4080.3
Human Labeled Utterances (hours)						
15.38	2.52	3.10	4.25	-	9.93	27.47
Large-Scale Data Pipeline Labeled Utterances (hours)						
155.50	34.11	27.81	99.92	145.09	1539.50	1733.50

Table 1. Training and evaluation data statistics.

4. EXPERIMENTAL EVALUATION

In this section, we benchmark each of the described systems in the previous section on two different evaluation datasets. We report our results in terms of per-class average precision (AP), and mean Average Precision (mAP) in percentage for classification task. For keyword detection, we use false accept rate, false reject rate, weighted AP, and F-1 score in percentage.

4.1. Training Protocol

We use the following training protocol for all the experiments to ensure equitable comparison and reproducibility. We use Adam optimizer with a learning rate of 4.0e-5, an epsilon value of 1e-8 and a weight decay factor of 0.2. A linear scheduler with a warm-up ratio of 0.1 is used. We use 8 A100 GPUs, with a per-device batch size of 32, and trained for up to 25 epochs. For evaluation, we use a 15 second chunk of audio at a time. For longer than 15 seconds audio segments, we use a stride of 15 sec to divide it into smaller chunks. We trained the models with 81 keywords and evaluated for the same set of keywords.

For the MTL model with task adapter, each task adapter has about 4.3M parameters. For our experiment, we did not investigate the optimal positions and the number of adapters to apply, and instead attach every transformer block with a task adapter for each task. This resulting in 26 total task adapters. Note that task adapter in encoder and information sharing in decoder are different design perspectives and are applied separately. Through our initial experiments, we observe that applying both together cause severe performance degeneration.

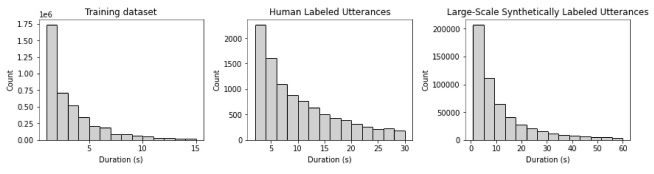

Fig. 3. Duration distributions of utterances in training and evaluation datasets.

Model Details		Toxicity Classification Results							Audio to Keyword Detection Results			
Model	Num. Params (M)	Profanity	Dating & Sexting	Racist	Bullying	Other	No Violation	mAP	False Accept Rate	False Reject Rate	Weighted AP	F1
Human Labeled Utterances												
Toxicity Classifier	94.6	98.12	54.40	**84.35**	**58.63**	-	78.04	**74.71**	-	-	-	-
Audio to Keyword	94.4	-	-	-	-	-	-	-	19.59	**2.51**	45.81	62.36
MTL	94.6	97.52	53.74	81.53	56.05	-	76.27	73.02	17.11	2.89	50.13	66.55
MTL + Info. Sharing	95.2	**98.15**	51.09	83.41	55.95	-	77.94	73.31	13.10	2.86	54.46	70.57
MTL + Task Adapter	207.2	96.76	**57.73**	80.74	54.53	-	**78.43**	73.64	**12.46**	3.27	**55.60**	**71.55**
Large-Scale Data Pipeline Labeled Utterances												
Toxicity Classifier	94.6	**69.03**	39.97	40.31	**55.35**	64.83	97.57	**61.18**	-	-	-	-
Audio to Keyword	94.4	-	-	-	-	-	-	-	5.41	**3.98**	32.25	54.30
MTL	94.6	66.17	37.61	39.54	52.06	62.37	97.46	59.20	4.45	4.70	36.76	58.27
MTL + Info. Sharing	95.2	67.49	36.46	35.04	52.07	**66.91**	97.62	59.27	4.00	4.81	38.82	59.65
MTL + Task Adapter	207.2	67.26	**40.09**	**42.32**	54.07	60.42	**97.68**	60.31	**3.76**	4.94	**39.94**	**60.53**

Table 2. Evaluation results of single task Toxicity Classification and Audio to Keyword Detection, and multi-task (MTL) architectures.

4.2. Training & Evaluation Datasets

We processed a large batch of audio data from Roblox's Voice Chat using the data pipeline (Section 2) to produce training labels (Table 1). The duration distribution of training data was limited to 15 seconds. We prepare two evaluation datasets, once is a more balanced human labeled dataset consisting of 9795 utterances, or about 27 hours of audio containing up to 30 seconds of audio per utterance, and a much larger evaluation set consisting of 537,311 utterances sampled from a real world distribution that are labelled using the data pipeline, comprising of about 1733 hours of audio data, containing up to 60 seconds of audio per utterance. Note that human labeled evaluations do not have an "Other" category, due to this category being a broad catch-all for a broad variety of toxic behavior captured by the text classifier. Finally, the duration distribution of the different datasets are shown in Figure 3, which motivates our choice of a 15 sec window for audio context in our pipeline.

4.3. Results

4.3.1. Toxicity Classification Results

We observe from Table 2 that for different distributions of evaluation data, the same model yields drastically different AP values for every class. This shows the difficulty of the toxicity classification task on real-world data distribution as such models generalize to thousands of hours of audio. We notice that the larger dataset with data pipeline labels is more discriminative across different model variants across different classes, and we hypothesize that the imbalanced distribution as well as the larger diversity of data in the larger dataset amplifies the performance differences between each model.

Profanity seems to be the best performing toxicity class to detect, and this makes sense because profanity requires shorter context windows than other toxicity categories to reliably detect it. Dating & Sexting and Racism can occur over longer sentences, as well as Bullying, and these differences do show up in our toxicity classification systems performance on each of these classes.

The comparison of per-class AP, with and without MTL, highlights some interesting nuances. We note that all multi-task variants exhibit a relative reduction in mAP values from the baseline, ranging from 1.4% to 3.2% across both datasets. The biggest changes in AP with abuse categories which require longer context windows, including Dating and Sexting, Racism and Bullying. Information sharing improves keyword specific categories such as profanity but drops for less keyword dependent categories such as Dating & Sexting and Racist classes. We also see that information sharing gives the best performance on the "Other" category over all other variants, which contains a lot of the toxic speech that are detected by keyword specific text classifiers. Adding a task adapter seems to alleviate the performance drop due to the keyword detection head by giving the classification task head its own set of parameters that can focus entirely on the classification. The task adapter and information sharing MTL variant results suggest that the toxicity classifier performance also depends on its ability to distinguish non-keyword based toxicity.

4.3.2. Audio to Keyword Detection Results

For keyword detection, we observe the impact of dataset distribution on key metrics is less pronounced between the balanced and real world distribution evaluation sets. Multi-task learning improves keyword detection across the board, with a relative F1 score improvement of around 6.3% to 11.5%, and a relative weighted AP improvement of around 9.4% to 23.8% across both datasets, with the largest gains seen by the information sharing and task adapter MTL variants. We note this is a significant improvement to the keyword detection task, and these architectures are viable for better explainable toxicity detection, with minimal loss in performance for toxicity classification. With only a 0.6% increase in parameters, information sharing improves upon the standard MTL pipeline with by up to 1-2% increase in weighted AP and F1 score at the cost of non-keyword based categories. The task adapter MTL model shows the most promise, getting the best audio to keyword performance with the smallest penalty to toxicity classification, particularly in categories that are not keyword-centric.

5. CONCLUSIONS

We present a novel multi-task learning architecture for audio-based toxicity detection, specifically toxicity classification and audio to keyword detection tasks. We benchmark the performance of proposed MTL models on large-scale real-world datasets, and find that toxicity classification helps improve audio to keyword classification, but such an inductive bias by the keyword task does not help all toxicity classes equally. The results presented here are for English, but the same model can be extended to multi-lingual toxicity detection and keyword detection. In the future, we plan to further optimize the model footprint for enabling deployment on compute resource constraint devices and continue to improve the classification performance for some of the challenging categories such as Dating & Sexting and Bullying. Overall, we believe this work would help accelerate the research community to pursue audio based solutions for toxicity detection.

6. REFERENCES

[1] Hugo Rosa, Nádia Pereira, Ricardo Ribeiro, Paula Costa Ferreira, Joao Paulo Carvalho, Sofia Oliveira, Luísa Coheur, Paula Paulino, AM Veiga Simão, and Isabel Trancoso, "Automatic cyberbullying detection: A systematic review," *Computers in Human Behavior*, vol. 93, pp. 333–345, 2019.

[2] Marcus Märtens, Siqi Shen, Alexandru Iosup, and Fernando Kuipers, "Toxicity detection in multiplayer online games," in *2015 International Workshop on Network and Systems Support for Games (NetGames)*. IEEE, 2015, pp. 1–6.

[3] Julian Risch and Ralf Krestel, "Toxic comment detection in online discussions," *Deep learning-based approaches for sentiment analysis*, pp. 85–109, 2020.

[4] Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee, "Deep learning models for multilingual hate speech detection," *arXiv preprint arXiv:2004.06465*, 2020.

[5] Pranav Malik, Aditi Aggrawal, and Dinesh K Vishwakarma, "Toxic speech detection using traditional machine learning models and bert and fasttext embedding with deep neural networks," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2021, pp. 1254–1259.

[6] Sreyan Ghosh, Samden Lepcha, S Sakshi, Rajiv Ratn Shah, and Srinivasan Umesh, "Detoxy: A large-scale multimodal dataset for toxicity classification in spoken utterances," *arXiv preprint arXiv:2110.07592*, 2021.

[7] Midia Yousefi and Dimitra Emmanouilidou, "Audio-based toxic language classification using self-attentive convolutional neural network," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 11–15.

[8] Wei-Cheng Lin and Dimitra Emmanouilidou, "Toxic speech and speech emotions: Investigations of audio-based modeling and intercorrelations," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 115–119.

[9] Ahlam Husni Abu Nada, Siddique Latif, and Junaid Qadir, "Lightweight toxicity detection in spoken language: A transformer-based approach for edge devices," *arXiv preprint arXiv:2304.11408*, 2023.

[10] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, pp. 335–359, 2008.

[11] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[12] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28492–28518.

[13] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[14] Quoc N. Le and Kip Kaehler, "How we scaled BERT to serve 1+ billion daily requests on CPUs," May 2020.

[15] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al., "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[16] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al., "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[17] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 4087–4091.

[18] Tara Sainath and Carolina Parada, "Convolutional neural networks for small-footprint keyword spotting," 2015.

[19] Ding Zhao, Tara N Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang, "Shallow-fusion end-to-end contextual biasing.," in *Interspeech*, 2019, pp. 1418–1422.

[20] Zoltán Tüske, Ralf Schlüter, and Hermann Ney, "Investigation on lstm recurrent n-gram language models for speech recognition," in *Interspeech*, 2018, pp. 3358–3362.

[21] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[22] Jonathan Pilault, Amine Elhattami, and Christopher Pal, "Conditionally adaptive multi-task learning: Improving transfer learning in nlp using fewer parameters & less data," *arXiv preprint arXiv:2009.09139*, 2020.

[23] Alex Kendall, Yarin Gal, and Roberto Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[24] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *International conference on machine learning*. PMLR, 2018, pp. 794–803.