USTED: IMPROVING ASR WITH A UNIFIED SPEECH AND TEXT ENCODER-DECODER

Bolaji Yusuf[1,2], Ankur Gandhe[1], Alex Sokolov[1]*

[1] Amazon Alexa, USA
[2] Boğaziçi University, Department of Electrical and Electronics Engineering, Istanbul, Turkey
[3] Brno University of Technology, Faculty of Information Technology, Speech@FIT, Czechia

ABSTRACT

Improving end-to-end speech recognition by incorporating external text data has been a longstanding research topic. There has been a recent focus on training E2E ASR models that get the performance benefits of external text data without incurring the extra cost of evaluating an external language model at inference time. In this work, we propose training ASR model jointly with a set of text-to-text auxiliary tasks with which it shares a decoder and parts of the encoder. When we jointly train ASR and masked language model with the 960-hour Librispeech and Opensubtitles data respectively, we observe WER reductions of 16% and 20% on test-other and test-clean respectively over an ASR-only baseline without any extra cost at inference time, and reductions of 6% and 8% compared to a stronger MUTE-L baseline which trains the decoder with the same text data as our model. We achieve further improvements when we train masked language model on Librispeech data or when we use machine translation as the auxiliary task, without significantly sacrificing performance on the task itself.

Index Terms— sequence-to-sequence, multitask, end-to-end ASR, masked language model, machine translation

1. INTRODUCTION

End-to-end (E2E) approaches to ASR arose as a more direct alternative to hybrid methods. By leveraging the considerable expressive power of large neural networks, E2E models obviate the need to train multiple disparate models while also considerably simplifying the onerous task of building an ASR decoder. This simplification, however, comes at a cost of data efficiency. Where the various modules of the hybrid approach can be estimated separately with various sources of data, doing the same for E2E models in a principled way is still an open problem.

Several works have attempted to improve E2E ASR by incorporating external data. Typically, this involves training with unpaired speech by using pseudo labels from an ASR system [1, 2, 3], pretraining with surrogate unsupervised objectives [4, 5, 6] or combining these into an ASR-TTS cycle consistency objective [7, 8].

Another line of work is concerned with incorporating unpaired text into the ASR model through fusion with an external language model (LM) [9, 10] or hallucinating corresponding speech to increase the pool of available ASR training data [11, 12, 13]. Recently, there has been a focus on manipulating the LM induced by the decoder of an E2E ASR model [14]. In [15, 16], the decoder LM score is subtracted before shallow fusion with an external LM to improve domain adaptation. In [17] on the other hand, the decoder is trained with a large text corpus alleviating the need for an external LM.

Inspired by T5 [18] which uses a multitask generative construction, in this paper, we take the view that ASR is not a problem in a vacuum but one of a set of interrelated sequence transduction tasks. Based on the observation that various sequence-to-sequence tasks can be framed as trying to generate text from their respective input modalities (e.g. ASR is the task of generating text from speech, MT is the task of generating text from text in another language etc.), we will use a set of sequence-to-sequence tasks with the same output language (English in our case) to train a single model which will generate the correct text irrespective of the modality of its input.

We propose the Unified Speech and Text Encoder-Decoder (USTED), an attention-based model trained on ASR along with a variety of text-to-text transduction tasks. Instead of a single speech encoder, USTED has a bank of shallow task-specific *modality* encoders which transform their respective inputs into a shared space from which a task-agnostic *context* encoder operates on them. Finally, a shared decoder autoregressively generates the desired text. The crux of our approach is sharing the parameters of the context encoder and the decoder across all tasks, a choice which we hypothesize will improve the ASR in two ways: we'll get the advantages of training the decoder on a larger text corpus since the text tasks contain a much larger number of training sentences than the ASR; similar to models pretrained on unlabeled speech, we'll get a better encoder by training the parameters on a surrogate objective (text-to-text). Moreover, from a wider perspective, we get a model capable of solving multiple tasks simultaneously, and benefits from doing so, without significantly increasing the number of parameters.

We conduct experiments on Librispeech and Opensubtitles which show that the proposed method offers an effective way of incorporating external text for improving ASR without requiring an external language model. Furthermore, we show masked language modeling to be a suitable text-to-text task for our purposes, thereby eliminating the need to search for text-to-text tasks with paired data.

2. MODEL

2.1. Attention-based encoder-decoder for ASR

The model proposed in this paper is based on the attention encoder-decoder (AED) paradigm for end-to-end ASR [19, 20]. An AED model with encoder parameters ϕ and decoder parameters θ directly models the distribution:

$$p_{\boldsymbol{\xi}}(\mathbf{y}|\mathbf{X}) = \prod_{u=1}^{U_y} p_{\boldsymbol{\xi}}(y_u|\mathbf{X}, y_{1:u-1}), \qquad (1)$$

where $\boldsymbol{\xi} = \{\phi, \theta\}$, $\mathbf{X} := (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \mathbb{R}^{N \times D_x}$ is a sequence of acoustic features and $\mathbf{y} := (y_1, \dots, y_{U_y})$ is the sequence of corre-
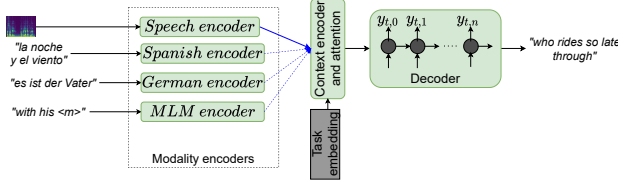

Fig. 1. Schema of the proposed approach. A bank of shallow modality encoder project an input feature into a space from which a shared encoder and decoder generate the corresponding textual output.

sponding transcripts in the form of words, characters, or character-based subword units.

The model's encoder transforms $\mathbf{X}$ into a hidden representation $\mathbf{H} := (\mathbf{h}_1, \ldots, \mathbf{h}_N) \in \mathbb{R}^{N \times D_h} = \phi(\mathbf{X})$. The decoder recursively models the distribution $p_{\boldsymbol{\theta}}(y_u | \mathbf{X}, y_{1:u-1})$ of the u-th token conditioned on the encoder output $\mathbf{H}$ and previously generated tokens $y_{1:u-1}$. At each decoding step, the encoder outputs are summarized into a context vector $\mathbf{c}_u^{\phi}(\mathbf{X}) = \sum_{n=1}^{N} w_{un} \mathbf{h}_n$, where the weights $\{w_{un}\}$ are computed with an additive attention mechanism [20]. The context vector is fed into the decoder along with a trainable embedding of the output at the previous step so that the conditional distribution is simplified to:

$$p_{\boldsymbol{\xi}}(\mathbf{y}|\mathbf{X}) = \prod_{u=1}^{U_y} p_{\boldsymbol{\theta}}(y_u | \mathbf{c}_u^{\phi}(\mathbf{X}), y_{u-1}). \tag{2}$$

The model is trained with the cross-entropy objective which minimizes the negative log-likelihood of the correct transcriptions $\hat{\mathbf{y}}$ of data $\mathbf{X}$ in a training set $\mathcal{D}$:

$$\mathcal{L}_{asr} = - \sum_{\mathbf{X}, \hat{\mathbf{y}} \in \mathcal{D}} \sum_{u=1}^{U_{\hat{y}}} \log p_{\boldsymbol{\theta}}(\hat{y}_u | \mathbf{c}_u^{\phi}, \hat{y}_{u-1}), \tag{3}$$

where the explicit dependence of $\mathbf{c}_u^{\phi}(\cdot)$ on $\mathbf{X}$ is henceforth dropped from the notation to reduce clutter.

2.2. Unified speech and text encoder-decoder

USTED, depicted in Figure 1, is a modified AED architecture which simultaneously operates on speech as well as a variety of text-to-text transduction tasks. Given a set of transduction tasks $\{T^1, \ldots, T^Q\}$ with associated data:

$$\mathcal{D} = \left\{ \mathcal{D}^1 = \{\mathbf{X}, \mathbf{y}\}^1, \mathcal{D}^2 = \{\mathbf{X}, \mathbf{y}\}^2, \ldots, \mathcal{D}^Q = \{\mathbf{X}, \mathbf{y}\}^Q \right\},$$

The model is composed of a bank of task-specific *modality* encoders $\{\phi^1, \ldots, \phi^Q\}$, a *context* encoder ϕ which is shared across tasks, and a single, task-agnostic decoder $\boldsymbol{\theta}$.

For a data sample $\mathbf{X}$ from the q-th task, the corresponding modality encoder transforms the input into a shared space from where it is further transformed by the shared encoder:

$$\tilde{\mathbf{H}} := (\tilde{\mathbf{h}}_1, \ldots, \tilde{\mathbf{h}}_N) = \phi^q(\mathbf{X})$$
$$\mathbf{H} = \phi(\tilde{\mathbf{H}}). \tag{4}$$

Note that for tasks with textual input, a trainable, task-specific, embedding layer is used to embed the discrete input symbols into a continuous space. Finally, as in regular AED, the decoder autoregressively generates textual symbols with context vectors $\mathbf{c}_u^{\phi, \phi^q}(\mathbf{X}) = \sum_{n=1}^{N} w_{un} \mathbf{h}_n$ computed from the encoder outputs.

2.2.1. Training method and objective

We train USTED with the same teacher-forced cross-entropy objective as the regular AED with the caveat that now we compute the loss for data across different tasks. Thus, we minimize the negative log-likelihood of generating the correct output sequences:

$$\mathcal{L}_{gen} = - \sum_{q=1}^{Q} \sum_{\mathbf{X}, \hat{\mathbf{y}} \in \mathcal{D}^q} \sum_{u=1}^{U_{\hat{y}}} \log p_{\boldsymbol{\theta}}(\hat{y}_u | \mathbf{c}_u^{\phi, \phi^q}, \hat{y}_{u-1}). \tag{5}$$

When training, we uniformly sample from the set of tasks and then sample data within the task. Thus, each batch only has data from a single task.

2.2.2. Task embedding

We experiment with cluing in the decoder (and shared encoder) on what task the current sample belongs to. We achieve this by prepending a trainable task-specific embedding, $\tilde{\mathbf{h}}_0^q$, to the output of the task specific encoder modifying (4) to:

$$\tilde{\mathbf{H}} := \left(\tilde{\mathbf{h}}_0^q, \tilde{\mathbf{h}}_1, \ldots, \tilde{\mathbf{h}}_N \right) = \left(\tilde{\mathbf{h}}_0^q, \phi^q(\mathbf{X}) \right). \tag{6}$$

We hypothesize that this would allow us to better generate text from the output distributions of the various tasks and we use it in all our experiments except where otherwise specified.

3. EXPERIMENTS

3.1. Experiment setup

3.1.1. Tasks and datasets

Speech recognition: We use the full 960 hours data from the Librispeech corpus [21] for training the ASR task. We use the various test splits from the same corpus for evaluation. In particular, we use the sum of word error rates (WER) on `dev-other` and `dev-clean` as the criterion for selecting best training checkpoints for the `test-clean`, `test-other` splits. We also evaluate on the `os-en-tts` set obtained by synthesizing speech from the 100k sentences of the English part of `test-es-en` described below with the TTS system described in [22].

Machine translation: As one of the side tasks, we explore MT, for which we take the paired Spanish-English (`es-en`) and German-English (`de-en`) sets from the Opensubtitles corpus [23] of aligned movie subtitles amounting to 61 million and 22 million sentences respectively. From each of these, we create splits of 100k sentences each (`test-es-en`, `test-de-en`) for evaluating MT performance, with the remaining data used for training. The training and evaluation splits are constructed such that no two splits contain sentences from the same movie. Note that, as indicated in Table 1, the contemporary, conversational language in the Opensubtitles data differs significantly from the literary, often archaic, forms

Table 1. Trigram perplexities of Librispeech and Opensubtitles language models evaluated on Librispeech and Opensubtitles dev sets.

LM training data	dev-clean	dev-other	test-es-*en*
Librispeech-960h	264	230	409
Librispeech-40m	175	163	329
Opensubtitles-es-*en*	476	426	131

Table 2. WER comparison between USTED and various baselines. Where applicable, the parenthesized R and K values denote the masking rate and the number of shared encoder layers respectively.

Model	Auxiliary dataset	Auxiliary task	dev-clean	dev-other	test-clean	test-other	os-en-tts
Baseline	-	-	5.6	14.2	5.9	15.2	18.1
MUTE-L [17]	Opensubtitles	-	4.9	12.7	5.1	13.5	14.7
USTED (K=3)	Opensubtitles	de-en MT	4.5	12.1	4.6	12.9	14.3
USTED (K=1)	Opensubtitles	**es-en MT**	**4.4**	**12.0**	**4.5**	**12.5**	**14.3**
USTED (K=1, R=0.4)	Opensubtitles	MLM	4.6	12.1	4.7	12.7	14.3
USTED (K=3, R=0.4)	Opensubtitles	MLM + MT	4.6	12.1	4.7	12.6	14.0
MUTE-L	Librispeech-40m	-	4.6	12.3	4.9	12.9	15.3
USTED (K=3, R=0.4)	Librispeech-40m	MLM	4.5	11.9	4.7	12.5	14.9

in Librispeech, reflected in the perplexity gaps between using the Librispeech LM on Opensubtitles dev set and vice versa.

Masked language modeling: The final training task that we explore is masked language modeling where we use the English part (61 million sentences) of the Spanish-English Opensubtitles. In this task, we corrupt a sentence by randomly masking some words and then requiring the model to reconstruct the *uncorrupted* sentence. Note that we reconstruct the entire sequence rather than just the masked tokens since the decoder is shared with other auto-regressive generative tasks and thus should be trained as a proper language model.

3.1.2. Model configuration

Our implementation of USTED is based on the listen, attend and spell architecture [19] with a 4-layer BiLSTM encoder and a two-layer LSTM decoder with four-headed additive attention. Each encoder layer has 1024 units in each direction. As described above, the decoder is shared across all tasks as is part of the encoder. Note that the 4 encoder layers include both the task-specific modality encoders (K layers) and the shared context encoder ($4 - K$ layers). For instance, when we share 3 layers, this means the shared context encoder has 3 BiLSTM layers and the modality encoders have 1 BiLSTM layers each. Thus, while the total number of parameters in the model reduces as the number of shared layers is increased, each task always has the same number of parameters available to it regardless of the number of shared layers.

The network outputs tokens from a unigram subword tokenizer [24] with 2500 tokens trained on Librispeech. For English-to-English text tasks, we use the same tokenizer for the input. For the translation tasks, we train a tokenizer with 2500 for the input language. The input text tokens are are embedded by task-specific embedding layers into 192 dimensional space. The input to the speech modality encoder are 64-dimensional log-filterbank features stacked with the two frames to the left and then downsasmpled by 3 resulting in a 192-dimensional input feature for every $30ms$.

3.1.3. Speech encoder pretraining

In preliminary experiments, we found that getting competitive ASR results required oversampling the ASR task. This however degraded performance on other tasks. We solve this issue by pretraining a standalone LAS ASR system whose encoder's parameters are transferred to initialize both the speech modality encoder and the context encoder. We randomly initialize the decoder and the other tasks' modality encoders as we did not observe any benefits in pretraining them. After pretraining, we sample training tasks uniformly, leading to faster convergence and better performance on auxiliary tasks.

3.2. Test set performance

Table 2 shows a comparison of USTED performance with two baselines: a standalone LAS model with no auxiliary tasks and MUTE-L which uses the text data to further train the decoder as described in [17]. All models use the Librispeech 960h for ASR training. We consider Opensubtitles and Librispeech as the source of auxiliary text data for both MUTE-L and USTED. Note that in the Opensubtitles setting, we use the English part of the Spanish-English translation corpus for training MUTE-L as well as MLM for USTED.

When we use Opensubtitles as the auxiliary dataset, we observe that all the variants of USTED improve upon MUTE-L which already outperforms the ASR-only baseline by a significant margin. When we use Spanish-English translation as the auxiliary task, we observe relative word error rate improvements of 9.8% and 5.9% on `test-clean` and `test-other` respectively. While using MLM as the auxiliary task for USTED results in more modest improvements, it has the advantage of only requiring the same unpaired text data as MUTE-L unlike machine translation task requiring paired data that is normally harder to obtain. Therefore, we'll conduct most of our further analyses on the MLM task.

Using Librispeech as the auxiliary dataset precludes us from using machine translation task since we have no paired data. USTED with MLM still outperforms MUTE-L in all test sets in this setting. We note that when compared to the corresponding USTED models trained with Opensubtitles auxiliary data, both models perform better on Librispeech test sets and worse on the Opensubtitles test sets; this supports the intuition that having a task from domain matching the test set leads to better performance.

Finally, we note that using Spanish-English translation as the auxiliary task achieves comparable performance with MLM and outperforms MUTE-L on Librispeech test sets even when the latter are trained with Librispeech text, while being significantly better than both on the synthesized Opensubtitles test set. This indicates that the performance gains realized from using a suitable task can make up for differences in output domain. However, finding such a suitable task *a priori* for any target test set remains an open question.

Table 3. WER of USTED as we vary the number of encoder layers that are shared by ASR and MLM.

Shared layers	0	1	2	3	4
dev-clean	5.0	4.7	4.9	4.7	4.8
dev-other	12.8	12.2	12.1	12.5	12.7

Table 4. WER as the MLM masking rate is varied.

Masking ratio	0	0.2	0.4	0.6	0.8	1
dev-clean	4.9	4.7	4.7	4.5	4.6	4.6
dev-other	12.5	12.5	12.2	12.2	12.3	12.2
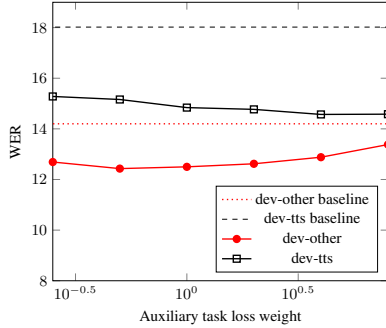
Fig. 2. WER on the `os-es-tts` set as the weight of the MLM training loss is varied.

3.3. MLM performance analysis.

As described in Section 2, in addition to sharing the decoder across all tasks, we also share the last few encoder layers. Table 3 shows the impact of changing varying the number of shared encoder layers from 0 (only share the decoder) to 4 (sharing the *entire* model) with the masking rate fixed to 0.4. We observe that sharing 0 layers performs worst with performance comparable to MUTE-L. The performance peaks with one shared encoder layer and starts to degrade as more layers are shared. This indicates that the gains obtained from shared information across tasks eventually get offset by the loss of capacity from reduction in total number of parameters.

Table 4 shows the effect of modifying the masking rate while keeping the number of shared encoder layers fixed to 1. The worst ASR performance is obtained when none of the input words are masked, i.e. having a sequence-to-sequence autoencoder as the auxiliary task. Performance improves with increasing masking rate with peak performance when 60% of the input text is masked with drop in performance afterwards. Surprisingly, even a masking rate of 1 outperforms the MUTE-L baseline despite having using the same information plus the length of the input. We hypothesize that even with all tokens masked in the input, sharing the encoder regularizes it and we will explore this further in future work

In Equation 5, each task is weighted equally and that is what we have used so far. Figure 2 shows the impact on ASR of re-weighting the losses of the MLM task while keeping the ASR loss weight at 1. On `dev-other`, we observe that the WER is best at 0.5 and starts to worsen as the weight is increased. We observed a similar trend on `dev-clean`, although we elected to exclude it from the figure to avoid warping the scale of the graph. On `dev-tts`, which is 24% of the entire `os-en-tts` set and is thus from the same domain as the MLM text, we observe that the WER improves as we increase the MLM weight up to a weight of 8 which is the maximum we tried.

3.4. Effect of task embedding

In all our experiments so far, we have prepended a task embedding to the output of each modality encoder. Tables 5 and 6 show the impact of the task embeddings on ASR and translation performance. While

Table 5. Impact of task embedding on ASR dev-set performance.

| | MLM | | MLM + MT | |
	clean	other	clean	other
USTED	4.8	12.2	4.6	12.1
- task embedding	4.8	12.3	4.7	12.2

Table 6. BLEU scores achieved on Spanish and German machine translation tasks.

	test-es-en	test-de-en
Baseline	31.6	26.3
USTED (MT)	31.5	25.8
USTED (MLM + MT)	29.6	24.2
- task embedding	23.1	18.5

Table 7. Examples of errors corrected by USTED. Errors made by the baseline are in red and corrections made by USTED are in blue.

System	Transcription
Baseline	THEY SAY ILLUMINATION BY CANNOLIDAYS THE PRETTIEST IN THE WORLD
USTED	THEY SAY ILLUMINATION BY CANDLE LIGHT IS THE PRETTIEST IN THE WORLD
Baseline	DON'T WORRY SAYS ODEIR IT'LL ALL COME RIGHT PRETTY SOON
USTED	DON'T WORRY SIZZLE DEAR IT'LL ALL COME RIGHT PRETTY SOON

the impact on ASR performance is minimal, the effect on MT is more drastic. Using the task embedding allows us to improve ASR without sacrificing much in terms of MT performance when compared to a baseline model trained on MT alone. Note that the performance of USTED on MT is on par with the standalone baseline when it is trained with two tasks, i.e. ASR and the relevant language pair.

3.5. ASR error qualitative analysis

Table 7 contains examples of some errors corrected by USTED. We observe that USTED reduces instances of acoustically plausible but linguistically nonsensical transcriptions, although we found that MUTE-L also makes similar corrections. While USTED fixes other kinds of errors, we could not find any conclusive patterns.

4. CONCLUSIONS

In this work, we've introduced a framework for jointly training ASR with text-to-text transduction tasks wherein parameters of an attention-based sequence to sequence model are shared across tasks. We have shown the efficacy of using machine translation and masked language modeling as the auxiliary tasks. Our model achieves significant improvements in ASR performance without incurring any computational overhead at inference time. Moreover, the added training cost is further justified by the fact that our multitask model also performs the auxiliary tasks, specifically machine translation, with minimal degradation in performance. We leave further exploration and analysis of auxiliary tasks from other domains and modalities as subject of future work.

5. REFERENCES

[1] Karel Veselý, Mirko Hannemann, and Lukáš Burget, "Semi-supervised training of deep neural networks," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 267–272.

[2] Jacob Kahn, Ann Lee, and Awni Hannun, "Self-training for end-to-end speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7084–7088.

[3] Sameer Khurana, Niko Moritz, Takaaki Hori, and Jonathan Le Roux, "Unsupervised domain adaptation for speech recognition via uncertainty driven self-training," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6553–6557.

[4] 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, 2020.

[5] Sameer Khurana et al., "A Convolutional Deep Markov Model for Unsupervised Speech Representation Learning," in *Proc. Interspeech 2020*, 2020, pp. 3790–3794.

[6] Andy T Liu, Shang-Wen Li, and Hung-yi Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2351–2366, 2021.

[7] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Listening while speaking: Speech chain by deep learning," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 301–308.

[8] Murali Karthick Baskar, Shinji Watanabe, Ramon Astudillo, Takaaki Hori, Lukáš Burget, and Jan Černocký, "Semi-Supervised Sequence-to-Sequence ASR Using Unpaired Speech and Text," in *Proc. Interspeech 2019*, 2019, pp. 3790–3794.

[9] Caglar Gulcehre et al., "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[10] Jan Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proc. Interspeech 2017*, 2017, pp. 523–527.

[11] Nick Rossenbach, Albert Zeyer, Ralf Schlüter, and Hermann Ney, "Generating synthetic audio data for attention-based speech recognition systems," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7069–7073.

[12] Gary Wang et al., "Improving speech recognition using consistent predictions on synthesized speech," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7029–7033.

[13] Murali Karthick Baskar et al., "Eat: Enhanced ASR-TTS for Self-Supervised Speech Recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6753–6757.

[14] Ehsan Variani, David Rybach, Cyril Allauzen, and Michael Riley, "Hybrid autoregressive transducer (hat)," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6139–6143.

[15] Erik McDermott, Hasim Sak, and Ehsan Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 434–441.

[16] Zhong Meng et al., "Internal language model training for domain-adaptive end-to-end speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7338–7342.

[17] Peidong Wang, Tara N. Sainath, and Ron J. Weiss, "Multitask Training with Text Data for End-to-End Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2566–2570.

[18] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[19] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[20] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[21] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[22] Amin Fazel, Wei Yang, Yulan Liu, Roberto Barra-Chicote, Yixiong Meng, Roland Maas, and Jasha Droppo, "SynthASR: Unlocking Synthetic Data for Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 896–900.

[23] Pierre Lison and Jörg Tiedemann, "OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles," in *Proceedings of LREC'16*, Portorož, Slovenia, May 2016, pp. 923–929, European Language Resources Association (ELRA).

[24] Taku Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 66–75.