ADAPTIVE DISCOUNTING OF IMPLICIT LANGUAGE MODELS IN RNN-TRANSDUCERS

Vinit Unni[1,†], *Shreya Khare*[2,†], *Ashish Mittal*[1,2], *Preethi Jyothi*[1], *Sunita Sarawagi*[1], *Samarth Bharadwaj*[2]

[1]Indian Institute of Technology Bombay, India
[2]IBM Research, India

ABSTRACT

RNN-Transducer (RNN-T) models have become synonymous with streaming end-to-end ASR systems. While they perform competitively on a number of evaluation categories, rare words pose a serious challenge to RNN-T models. One main reason for the degradation in performance on rare words is that the language model (LM) internal to RNN-Ts can become overconfident and lead to hallucinated predictions that are acoustically inconsistent with the underlying speech. To address this issue, we propose a lightweight adaptive LM discounting technique ADAPTLMD, that can be used with any RNN-T architecture without requiring any external resources or additional parameters. ADAPTLMD uses a two-pronged approach: 1. Randomly mask the prediction network output to encourage the RNN-T to not be overly reliant on it's outputs. 2. Dynamically choose when to discount the implicit LM (ILM) based on rarity of recently predicted tokens and divergence between ILM and implicit acoustic model (IAM) scores. Comparing ADAPTLMD to a competitive RNN-T baseline, we obtain up to 4% and 14% relative reductions in overall WER and rare word PER, respectively, on a conversational, code-mixed Hindi-English ASR task.

***Index Terms*—** RNN-Transducer, Implicit Language Model, Rare Word ASR

1. INTRODUCTION

End-to-end RNN Transducer (RNN-T) models [1] are rapidly becoming the de facto model for streaming speech recognition systems. RNN-T models consist of two independent encoders: an acoustic encoder that processes an acoustic signal one frame at a time, and, a language encoder consisting of an auto-regressive model that encodes previous tokens and acts as an implicit language model. Outputs from the acoustic and language encoders are combined and input to a joint network that makes the final RNN-T predictions. Such an architecture conveniently integrates training of the language model along with the acoustic model, and enables streaming ASR.

While RNN-T models perform competitively on various benchmarks [2, 3], they struggle with rare word predictions [4]. For words that rarely appear in the training data, the language encoder of the RNN-T could lead the network to favour hypotheses that are unfaithful to the underlying speech. This problem is compounded when dealing with out-of-domain ASR evaluations where the test speech is drawn from domains unseen during training. For example, our baseline RNN-T system hallucinated "canara bank" as "amazon" when trained on retail data but tested on banking data. More such cases of gross overriding of the acoustic input by LM hallucinations appear in Table 3.

Recent work on RNN-T models have used LM fusion techniques to improve rare word ASR predictions [5, 6, 4] but have typically relied on external resources (raw text, lexicons, etc.). In this work, we adopt a more direct approach by focusing on the RNN-T model itself and proposing model changes that aim at arresting an overconfident implicit LM from making predictions that are at odds with the underlying speech. Our proposed technique ADAPTLMD has the following two main features:

- We randomly mask outputs from the language encoder to the joint network, thus making the RNN-T model more robust to its spurious outputs.
- We propose a new dynamic discounting scheme that identifies regions of interest by examining discrepancies between the acoustic and language encoders and computing the rarity of recent tokens in the predicted hypothesis.

On out-of-domain evaluations, we observe statistically significant reductions in overall WER (4% relative) and large relative PER improvements of up to 14% on rare words.

Related Work. Prior work has handled rare word prediction and out-of-vocabulary detection [7, 8, 9] by modeling at the subword level using hybrid LMs [10, 11, 12]. More recent work on end-to-end models has mainly tackled the problem of out-of-domain tokens using techniques such as shallow fusion [13] where hypotheses are rescored using an external LM trained on text in the target domain. Many variants of shallow fusion use some form of contextual LMs [14, 4, 15, 16] that focus on fixing words appearing in specific contexts (e.g., involving named entities). A popular extension of shallow fusion, routinely employed in RNN-T models, is the density ratio technique [17] that additionally discounts the scores using an external LM trained on text from the training domains. This technique has seen further improvements by discounting

† Equal contribution.

using an implicit LM instead of an external LM [18, 19]. Both shallow fusion and the density ratio approach are inference-based techniques and hence preferred more than other techniques like deep fusion [13] and cold fusion [5] that require retraining with the new LMs. Besides LM rescoring, other techniques addressing tail performance include the use of sub-word regularization [15], using data augmentation via pronunciation dictionaries [15, 16, 20] or synthetic samples [21] and finetuning the prediction network using target text [22]. In concurrent work, [23] tries to limit LM overtraining by combining the two encoded representations using a gate and using a regularizer to slow the training of the LM. In contrast to most of the above-mentioned prior work, our technique fits within existing RNN-T architectures without requiring any additional parameter training or requiring external resources.

2. BACKGROUND: RNN-TRANSDUCERS

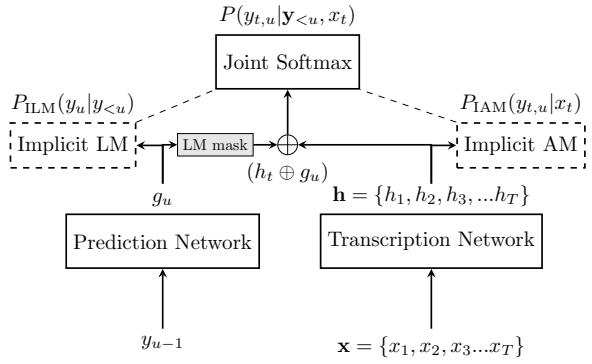
Consider an acoustic input $\mathbf{x} = \{x_1, x_2, x_3...x_T\}$ with corresponding output text transcript $\mathbf{y} = \{y_1, y_2, y_3...y_K\}$ where each $y_j \in \mathcal{V}$, the output vocabulary. $\mathcal{V}$ also includes a special blank token ϵ. An RNN-T model comprises a Prediction Network (PN), a Transcription Network (TN) and a joint network (Figure 1). The PN takes as input the previously emitted *non-blank* tokens $\mathbf{y}_{<u} = y_1 \dots y_{u-1}$, and auto-regressively generates an encoding for the next output token as g_u. The TN takes as input the acoustic signal $\mathbf{x}$ and outputs encoder states for each of the time steps $\mathbf{h} = \{h_1, h_2, h_3, ...h_T\}$. Henceforth, we will also use implicit acoustic model (IAM) and implicit LM (ILM) to refer to the TN and PN, respectively. The joint network J for each time step $t \leq T$ and token position $u \leq K$ takes as inputs the encoded vectors h_t and g_u and generates a probability distribution over the vocabulary $\mathcal{V}$ as:

$$\begin{aligned} P_{\text{rnnt}}(y_{t,u}|\mathbf{y}_{<u}, x_t) = &\quad \text{softmax}\{J(h_t \oplus g_u)\} \\ g_u = \text{PN}(\mathbf{y}_{<u}), &\qquad h_t = \text{TN}(\mathbf{x}, t) \end{aligned} \tag{1}$$

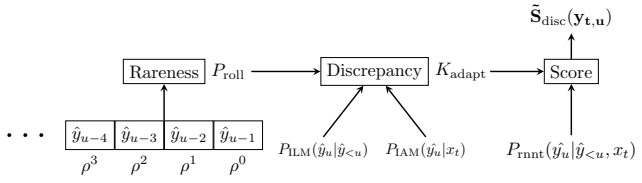
Traditionally, the $\oplus$ is an addition operation, however, other operations can be used too. J is any feed-forward network, PN is typically an RNN and TN a Transformer. The output probability P_{rnnt} over the $[t, u]$ space are marginalized during training to maximize the likelihood of the token sequence $y_1, \dots, y_K$ using an efficient dynamic programming algorithm. During decoding, beam-search is used to find the best possible $[t, u]$ alignment and output token sequence [1].

3. OUR APPROACH

A limitation of the symmetric integration of the language encoding (g_u) with the acoustic encoding (h_t) is a systematic bias against rare words, often resulting in hallucination of words that have no connection with the underlying speech. We seek to temper such overt influence of the ILM when we expect its predictions to be noisy. Our main challenge is how



(a) Training: LM masking and implicit LM+AM loss



(b) Decoding: Adaptive discounting of implicit LM

Fig. 1: Overview of ADAPTLMD

to detect, during online decoding, if the ILM is misleading. We propose a light-weight method called ADAPTLMD that can be applied on existing RNN-T architectures without any external lexicons or additional parameters.

ADAPTLMD differs from existing RNN-T based ASR in two ways. First, during training we impose random masking of the ILM output g_u, so that the joint network learns to be robust to spurious embeddings by the ILM. Second, during decoding we impose an *adaptive* discounting of the ILM on the output distribution P_{rnnt}. We dynamically choose when to discount based on two factors: (1) The discrepancy between the IAM and ILM signals, and (2) The rarity of recent tokens in the current hypothesis. We compute these terms using existing RNN-T modules and describe these in Sections 3.1 and 3.2 respectively. Our formula for adaptive ILM discounting based on these two scores, and the overall decoding algorithm is described in Section 3.3. The final training algorithm of ADAPTLMD appears in Section 3.4.

3.1. Discrepancy between LM and AM

The RNN-T architecture allows easy extraction of the independent token distribution of the TN and PN components by masking the h_t and g_u, respectively. We call these P_{ILM} and P_{IAM} and define as:

$$P_{\text{IAM}}(y_t|\mathbf{x}, t) = \quad \text{softmax}\{J(h_t \oplus \mathbf{0})\} \tag{2}$$

$$P_{\text{ILM}}(y_t|\mathbf{y}_{<u}) = \quad \text{softmax}\{J(\mathbf{0} \oplus g_u)\} \tag{3}$$

where $\mathbf{0}$ denotes an identity vector. These distributions are trained along with the primary P_{rnnt} as elaborated in Sec 3.4.

The distance between $P_{\text{ILM}}, P_{\text{IAM}}$ can be any divergence metric. We use *KL-divergence*:

$$\text{D}(P_{\text{ILM}}||P_{\text{IAM}}) = \sum_{y \in \mathcal{V}} P_{\text{ILM}}(y|\mathbf{y}_{<u}) \log \frac{P_{\text{ILM}}(y|\mathbf{y}_{<u})}{P_{\text{IAM}}(y|x_t)} \quad (4)$$

.

3.2. Rarity of Recent Tokens

We enable discounting at rare locations within a currently decoded hypothesis. Instead of external resources we depend on the implicit LM P_{ILM} to characterize rare regions. Let the current hypothesis be $\mathbf{y} = \{y_1, y_2, y_3, ...y_{u-1}\}$. We define a quantity $P_{\text{roll}}(\mathbf{y})$ that indicates a rolling average of the ILM probabilities over the recent tokens in the hypothesis $\mathbf{y}$. We choose a $0 \leq \rho \leq 1$ to compute $P_{\text{roll}}(\mathbf{y} + y_u)$ of $\mathbf{y}$ extended by a new token y_u at step u incrementally from the $P_{\text{roll}}(\mathbf{y})$ as:

$$P_{\text{roll}}(\mathbf{y} + y_u) = \rho P_{\text{roll}}(\mathbf{y}) + P_{\text{ILM}}(y_u) \quad (5)$$

As we want the discounting to be higher for rarer words where the implicit LM confidence on the hypothesis generated so far is low, we use $1 - P_{\text{roll}}(\mathbf{y})$ as a measure of the local rarity of recent tokens of a current hypothesis $\mathbf{y}$.

3.3. Decoding with Implicit LM Discounting

Using the above two scores, we adaptively discount the influence of the ILM in regions where the it is less confident and where there is discrepancy between the ILM and IAM as follows: Let $\mathbf{y} = y_1, \dots y_{u-1}$ denote the current hypothesis on the beam. The discounted score of the next token $y_{t,u}$ by which we extend $\mathbf{y}$ at position u from frame t is then computed as follows:

$$\tilde{S}_{\text{disc}}(y_{t,u}|\mathbf{y}, x_t) = \log P_{\text{rnnt}}(y_{t,u}|\mathbf{y}, x_t) - \lambda \max(0, D_{\text{adapt}}(y_{t,u}, \mathbf{y})) \log P_{\text{ILM}}(y_u|\mathbf{y})$$

$$D_{\text{adapt}}(y, \mathbf{y}) = \begin{cases} (1 - P_{\text{roll}}(\mathbf{y}))\text{D}(P_{\text{ILM}}(y)||P_{\text{IAM}}(y)) & \text{if } y \neq \epsilon \\ 0 & \text{else.} \end{cases}$$

where λ is a tunable parameter. While decoding we use the above discounted scores $\tilde{S}_{\text{disc}}$ as a drop-in replacement for the original $\log P_{\text{rnnt}}$ scores in the beam-search algorithm. With this discounted score, we defer to predictions of the implicit AM and discount the predictions by the P_{ILM} in regions of low confidence (indicated by P_{roll}). The only additional state we need to maintain for each hypothesis on the beam is P_{roll}. The rest of the inference algorithm stays unchanged. We extended the *Alignment-Length Asynchronous Decoding (ALSD)* [24] with the above changes as the decoding algorithm for ADAPTLMD.

3.4. Overall Training Objective

Our training strategy differs from RNN-T's in two ways: (1) We use random masking of the PN encoding to encourage the RNN-T not to be overly reliant on its output, and (2) we introduce extra losses for training the P_{ILM} and P_{IAM}.

$$\begin{aligned} &\log \sum_{\text{align t}} P_{\text{rnnt}}(y_{t,u}|h_t \oplus \text{maskedLM}_\eta(g_u)) \\ &+\alpha \log \sum_{\text{align t}} P_{\text{ILM}}(y_u|\mathbf{0} \oplus g_u) \\ &+\beta \log \sum_{\text{align t}} P_{\text{IAM}}(y_{t,u}|h_t \oplus \mathbf{0}) \end{aligned} \quad (6)$$

In eqn (6), $\text{maskedLM}_\eta(g_u)$ denotes the random masking of the TN where with a probability η we replace g_u with $\mathbf{0}$.

4. EXPERIMENTS AND RESULTS

Datasets. We conduct experiments on a proprietary dataset consisting of code-mixed speech (Hi-En) in a call-centre setting. This dataset contains 628 hours of speech and is divided into four domains: Banking (184.87 hrs), Insurance (165.08 hrs), Retail (135.87 hrs) and Telco (142.33 hrs). Domain-wise durations of train/test splis can be found in Table 1.

Architecture. We use a baseline RNN-T model where the transcription network consists of 6 Conformer [25] encoder layers with an internal representation size of 640 and the prediction network is a single layer LSTM with a 768 dimensional hidden state. The conformer consists of 8 attention heads of dim 27 and has a kernel size of 31 for the convolutional layers. The output of the transcription network is projected to a 256 dim vector h_t. The LSTM hidden state outputs a 10 dim embedding which is further projected to a 256 dim vector g_u. The joint network consists of an additive operation between h_t and g_u followed by a $\tanh$ nonlinearity. This is projected to the final layer of size 123 that corresponds to the size of our character-based vocabulary $\mathcal{V}$. We use the *Noam* optimizer [26] for training with a learning

Train (Hours) / Test (Hours)	ASR System	WER	CER	PER	Rare PER	Rare CER
All-train (477 hrs) / All-test (75 hrs)	RNNT	14.5	13.5	12.0	57.7	71.2
	ILMT	14.5	13.4	12.1	53.8	66.6
	ADAPTLMD	**14.1**	**13.1**	**11.8**	**52.2**	**63.9**
Telco + Retail + Insurance (327 hrs) / Test (58 hrs)	RNN_T	14.5	13.6	12.3	66.9	68.2
	ILMT	14.6	13.4	12.0	69.3	70.5
	ADAPTLMD	**14.2**	**13.1**	**11.9**	**63.2**	**59.6**
Banking + Retail + Insurance (374 hrs) / Test (55 hrs)	RNN_T	15.2	14.0	12.7	68.0	76.6
	ILMT	15.9	14.5	**12.3**	63.8	74.2
	ADAPTLMD	**14.9**	**13.6**	**12.3**	**60.2**	**63.3**
Banking + Telco + Insurance (381 hrs) / Test (55 hrs)	RNN_T	14.2	13.2	12.0	65.3	72.7
	ILMT	14.1	13.8	11.8	64.6	65.7
	ADAPTLMD	**13.8**	**12.9**	**11.6**	**63.1**	**63.8**
Banking + Telco + Retail (349 hrs) / Test (57 hrs)	RNN_T	15.0	14.4	13.1	66.9	68.2
	ILMT	15.9	14.5	13.2	69.3	70.5
	ADAPTLMD	**15.1**	**14.0**	**12.7**	**63.5**	**59.6**

Table 1: In-domain ASR Results.

Test Data	ASR-System	WER	CER	PER	Rare PER	Rare CER
Test-Banking (110 hrs)	RNN-T	22.4	20.0	18.5	70.8	75.9
	D.R.	22.2	19.9	18.4	70.8	74.6
	ILMT	22.4	20.0	18.5	70.9	75.9
	ADAPTLMD	**21.5**	**18.7**	**17.1**	**67.4**	**71.8**
Test-Insurance (95 hrs)	RNN-T	18.4	17.7	16.1	70.2	76.7
	D.R.	18.1	17.7	16.0	69.1	75.6
	ILMT	18.7	17.6	15.9	65.7	72.2
	ADAPTLMD	**17.7**	**16.4**	**14.8**	**60.7**	**67.2**
Test-Retail (71 hrs)	RNN-T	24.7	22.4	20.6	76.7	81.3
	D.R.	24.6	22.1	20.4	76.0	81.3
	ILMT	24.5	21.4	19.7	72.4	**77.5**
	ADAPTLMD	**23.7**	**21.2**	**19.5**	**72.2**	78.3
Test-Telco (75 hrs)	RNN-T	19.4	18.6	17.1	68.3	72.6
	D.R.	19.1	18.2	16.8	70.5	73.7
	ILMT	19.3	18.1	16.4	**63.8**	68.9
	ADAPTLMD	**18.6**	**17.5**	**16.0**	64.3	**68.6**

Table 2: Out-domain ASR Results.

rate of 0.0001. LM masking while training is performed with a probability of $\eta = 0.2$ and weights for the implicit LM and AM are $\alpha = \beta = 0.125$. The hyperparamters λ and ρ are chosen by tuning on a held-out set.

Baselines. Our in-domain results are compared against a vanilla RNN-T and an ILMT model trained with an implicit LM [27]. For out-of-domain experiments, we also compare with the density ratio approach [17] using external RNN-LMs (1024 units, 1 layer) trained on the source and target datasets.

Results. We show WERs/PERs/CERs for both in-domain and out-of-domain settings, where test utterances are drawn from domains that are either seen or unseen during training, respectively. Along with evaluations on the entire test set in each setting, we also separately compute PERs and CERs for rare words that appear less than 20 times during training.[1]

Table 1 shows in-domain results from training on speech from all domains, and four other training settings where we leave out one domain at a time. ADAPTLMD outperforms both RNN-T and ILMT in terms of WER reductions in almost all settings. Reductions in PER using ADAPTLMD for the rare words are much more significant. Table 2 shows out-of-domain results on four test domains that were each held out during training. Relative to the in-domain setting, ADAPTLMD outperforms all the baselines on overall WERs by a statistically significant margin (at $p < 0.001$ using the MAPSSWE test [28]). ADAPTLMD gives large PER improvements on rare words for all four test domains. These PER improvements predominantly stem from fixing the errors introduced by LM hallucinations of the RNN-T baseline. Table 3 provides a few examples of such hallucinations by the RNN-T baseline where the predictions are at times completely inconsistent with the underlying acoustics (e.g., *business→discount*, *call→contract*, etc.). We also show examples of ADAPTLMD predictions that may not be identical to the ground truth but are acoustically faithful, unlike the RNN-T baseline (e.g., *ikatees→iktees*, *vicky→vikkee*, etc.).

[1]We do not report rare word WERs since they were 100% or higher.

Groundtruth	*Ki aap apanaa* business (bɪznəs) account *kab* open (əʊpən) *karvana*
RNN-T	*ki aap apanaa* discount (dɪskaʊnt) account *kab bund (bənd)* karvana
ADAPTLMD	*ki aap apanaa* business (bɪznəs) account *kab* open (əʊpən) karvana
Groundtruth	*jisakaa pataa hain* plot (plɒt) number *ikatees (iktiːs) a..*
RNN-T	*jisakaa pataa hai* block (blɒk) number *tees (tiːs) ek*
ADAPTLMD	*jisakaa pataa hai* plot (plɒt) number *iktees (ikˈt̪ iːs) e..*
Groundtruth	*..ab aapki* call (kɔl) transfer *kar dee jaayegi..*
RNN-T	*..ab aapki* contract (kɔntɪ ækt) transfer *kar dee jaayegi..*
ADAPTLMD	*..ab aapki* call (kɔl) transfer *kar dee jaayegi..*
Groundtruth	*..mera pooraa naam hai..* vicky rajak (vikki ɪəʤək)..
RNN-T	*..mera pooraa naam hai..* reeti raagav (riːti raːgəv)..
ADAPTLMD	*..mera pooraa naam hai ..* vikkee raaj (vikkiː raːʤ)..
Groundtruth	*..ye online* activate (æktɪvejt) *karavaa sakataa kee naheen*
RNN-T	*..mujhe online* network (nɛtwɪ ķ) *karavaa sakataa hoon*
ADAPTLMD	*..ye online* activate (æktɪvejt) *karavaa sakataa hoon..*
Groundtruth	*namaskaar sir main apane* loan (lown) *ko fir se*
RNN-T	*namaskaar sir main apane* block (blɒk) *ko fir se*
ADAPTLMD	*namaskaar sir main apane* loan (lown) *ko fir se*

Table 3: Anecdotes illustrating LM hallucinations. Italicized text is Romanized Hindi. Tokens of interest are in blue.

Table 4 shows an ablation analysis of ADAPTLMD. We compare a baseline RNN-T with variants containing P_{ILM} and P_{IAM} losses (as in eqn (6)) and TN masking. Discounting is particularly beneficial for rare word predictions.[2]

	WER	CER	PER	Rare PER	Rare CER
RNN-T	14.5	13.5	12.0	57.7	71.2
RNN-T + P_{ILM} + P_{IAM}	14.5	13.2	11.9	53.4	66.5
RNN-T+P_{ILM} +P_{IAM} + TN mask	14.4	**13.1**	**11.8**	53.1	65.9
ADAPTLMD	**14.1**	**13.1**	**11.8**	**52.2**	**63.9**

Table 4: Ablations study to demonstrate the effect of TN masking and implicit LM/AM training in ADAPTLMD.

5. CONCLUSION

We propose ADAPTLMD, a new adaptive LM discounting technique that is dynamically invoked and can be used within any RNN-T model without requiring any additional resources. This technique particularly benefits rare words for which the LM internal to an RNN-T can generate predictions that do not match the underlying audio. ADAPTLMD yields consistent performance improvements on overall WERs and rare word predictions in both in-domain and out-of-domain evaluation settings for code-mixed Hindi-English ASR tasks across four domains.

6. ACKNOWLEDGEMENTS

The authors from IIT Bombay gratefully acknowledge support from IBM Research, specifically the IBM AI Horizon Networks-IIT Bombay initiative.

[2]Discounting the entire utterance with a static discount factor, instead of our adaptive scheme in ADAPTLMD, worsens overall WERs.

7. REFERENCES

[1] Alex Graves, "Sequence Transduction with Recurrent Neural Networks," 2012.

[2] Kanishka Rao, Haşim Sak, and Rohit Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *ASRU-2017*.

[3] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al., "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019*.

[4] Vijay Ravi, Yile Gu, Ankur Gandhe, Ariya Rastrow, Linda Liu, Denis Filimonov, Scott Novotney, and Ivan Bulyko, "Improving accuracy of rare words for rnn-transducer through unigram shallow fusion," 2020.

[5] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates, "Cold fusion: Training seq2seq models together with language models," 2017.

[6] Cal Peyser, Sepand Mavandadi, Tara N Sainath, James Apfel, Ruoming Pang, and Shankar Kumar, "Improving tail performance of a deliberation e2e asr model using a large text corpus," 2020.

[7] Carolina Parada, Mark Dredze, Denis Filimonov, and Frederick Jelinek, "Contextual information improves oov detection in speech," 2010.

[8] Hui Lin, Jeff Bilmes, Dimitra Vergyri, and Katrin Kirchhoff, "Oov detection by joint word/phone lattice alignment," in *ASRU-2007*.

[9] Ariya Rastrow, Abhinav Sethy, and Bhuvana Ramabhadran, "A new method for oov detection using hybrid word/fragment system," in *ICASSP-2009*.

[10] Maximilian Bisani and Hermann Ney, "Open vocabulary speech recognition with flat hybrid models," in *Interspeech-2005*.

[11] A. Yazgan and M. Saraclar, "Hybrid language models for out of vocabulary word detection in large vocabulary conversational speech recognition," in *ICASSP-2004*.

[12] Jinyu Li, Guoli Ye, Rui Zhao, Jasha Droppo, and Yifan Gong, "Acoustic-to-word model without oov," 2017.

[13] Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "On using monolingual corpora in neural machine translation," 2015.

[14] Alex Gruenstein et al., "An efficient streaming non-recurrent on-device end-to-end model with improvements to rare-word modeling," 2021.

[15] Duc Le, Gil Keren, Julian Chan, Jay Mahadeokar, Christian Fuegen, and Michael L. Seltzer, "Deep shallow fusion for rnn-t personalization," 2020.

[16] 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*.

[17] Erik McDermott, Hasim Sak, and Ehsan Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," 2020.

[18] Ehsan Variani, David Rybach, Cyril Allauzen, and Michael Riley, "Hybrid autoregressive transducer (hat)," 2020.

[19] Zhong Meng et al., "Internal language model estimation for domain-adaptive end-to-end speech recognition," 2020.

[20] Norihide Kitaoka, Bohan Chen, and Yuya Obashi, "Dynamic out-of-vocabulary word registration to language model for speech recognition," .

[21] Jinyu Li et al., "Developing rnn-t models surpassing high-performance hybrid models with customization capability," 2020.

[22] Janne Pylkkönen, Antti Ukkonen, Juho Kilpikoski, Samu Tamminen, and Hannes Heikinheimo, "Fast text-only domain adaptation of rnn-transducer prediction network," 2021.

[23] Chao Zhang, Bo Li, Zhiyun Lu, Tara N. Sainath, and Shuo yiin Chang, "Improving the fusion of acoustic and text representations in rnn-t," in *ICASSP-2022*.

[24] George Saon, Zoltán Tüske, and Kartik Audhkhasi, "Alignment-length synchronous decoding for rnn transducer," in *ICASSP-2020*.

[25] Anmol Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," 2020.

[26] Ashish Vaswani et al., "Attention is all you need," 2017.

[27] Zhong Meng, Naoyuki Kanda, Yashesh Gaur, Sarangarajan Parthasarathy, Eric Sun, Liang Lu, Xie Chen, Jinyu Li, and Yifan Gong, "Internal language model training for domain-adaptive end-to-end speech recognition," 2021.

[28] L. Gillick and S.J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *ICASSP-1989*.