INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Understanding Medical Conversations: Rich Transcription, Confidence Scores & Information Extraction

Hagen Soltau, Mingqiu Wang, Izhak Shafran, Laurent El Shafey

Google, USA
{soltau, mingqiuwang, izhak, shafey}@google.com

Abstract

In this paper, we describe novel components for extracting clinically relevant information from medical conversations which will be available as Google APIs. We describe a transformer-based Recurrent Neural Network Transducer (RNN-T) model tailored for long-form audio, which can produce rich transcriptions including speaker segmentation, speaker role labeling, punctuation and capitalization. On a representative test set, we compare performance of RNN-T models with different encoders, units and streaming constraints. Our transformer-based streaming model performs at about 20% WER on the ASR task, 6% WDER on the diarization task, 43% SER on periods, 52% SER on commas, 43% SER on question marks and 30% SER on capitalization. Our recognizer is paired with a confidence model that utilizes both acoustic and lexical features from the recognizer. The model performs at about 0.37 NCE. Finally, we describe a RNN-T based tagging model. The performance of the model depends on the ontologies, with F-scores of 0.90 for medications, 0.76 for symptoms, 0.75 for conditions, 0.76 for diagnosis, and 0.61 for treatments. While there is still room for improvement, our results suggest that these models are sufficiently accurate for practical applications.

1. Introduction

Medical conversations are an important source of information for clinical documentation and decision making. As the automatic speech recognition (ASR) and natural language processing (NLP) technology has matured, there has been considerable research interest in automatically extracting useful information from these conversations [1, 2, 3]. Previous work has focused on certain sub-problems such as summarization from transcripts, error analysis of ASR, or adapting tools from text to ASR transcripts. Instead, we focus on the minimal components necessary for extracting clinical information from raw audio, starting with a rich transcription of the medical conversation, as described in Section 2. We delve into estimation of confidence scores in Section 3, which is one of the most requested features in a medical ASR so users can decide when to trust the correctness of the recognized outputs. Finally, we describe a Recurrent Neural Network Transducer (RNN-T) based tagger in Section 4 which is the first of its kind to our knowledge.

2. Rich Transcription

Speech recognition is the first step to understand medical conversations. Before delving into the details of our ASR models, it is useful to understand the nature of medical conversations and the associated challenges.

The audio in medical conversations is substantially different from applications such as medical dictations or digital assistant, in terms of speakers, speaking styles and duration. The conversations may involve 2-5 speakers – doctor, nurse, patient, caregiver and other clinical support staff. The audio is typically captured by a far-field microphone, with a large variance in the distance from the speakers. In contrast to other conversational corpora (e.g. Switchboard), the style is collaborative, with relatively low-cross talk, as all parties have vested interest in addressing the issue at hand efficiently and effectively. More importantly, the conversations are fairly long. While the average duration is about 11 minutes, some conversations exceed more than an hour. Note, several recent papers treat long-form audio, for example on LibriSpeech task, as concatenation of short segments, which results in sub-optimal performance. For more details on the corpus, see [4].

Since the overall goal is to understand medical conversations, we need to create a *rich* transcription of the conversations that include not only punctuation, capitalization and confidence scores but also perform the diarization task that involves inferring speaker turns and associating them with the speakers so we know who spoke what and when. Since our task is akin to the NIST Rich Transcription tasks, we borrowed confidence and diarization metrics from them [5]. In the following, we discuss how these requirements impact modeling choices.

The factorization of the speech recognition models into separately trained acoustic $P(X|W)$ and language models $P(W)$, first formulated as Hidden Markov Model (HMM), continues to be a dominant choice for ASR. In this framework, there is a lack of strong coupling between acoustic and language models even when the acoustic models are trained with discriminative functions such as MMIE or MPE [6, 7] or the acoustic models are modeled by LSTMs [8] or transformers [9, 10]. This lack of coupling makes it difficult to utilize both acoustic and linguistic cues simultaneously in rich transcription.

Alternative approaches based on *sequence-to-sequence models with an attention mechanism* do jointly optimize the acoustic (encoder) and the language (decoder) components [11, 12, 13]. While these are apt models for rich transcription, their output suffers from deletions and repetitions, particularly when the input sequence is long. This has been somewhat alleviated in practice by *ad hoc* methods such as input-coverage penalties and monotonic alignment constraints. In previous work on medical conversations, the deletion issue went unnoticed because the models were trained and evaluated on selected segmented data where difficult to recognize segments were excluded [14].

Recurrent Neural Network Transducer (RNN-T) models are similar to attention-based sequence-to-sequence models in that the encoder and the decoder are optimized jointly with a single cost function [15], making these models also apt for rich transcription. Furthermore, the inference is performed in a time-synchronous fashion with a beam search process that accounts for *every* input frame. So, unlike the attention model, they do not suffer from the deletion or duplication problems and they are also well-suited for streaming applications [16].

http://dx.doi.org/10.21437/Interspeech.2021-691

In our previous work, we demonstrated that RNN-Ts can accurately predict speaker turns and two speaker role labels in a non-streaming application using bidirectional LSTM encoders [4]. In this paper, we investigate how RNN-T models can be applied in a streaming ASR application for rich transcription of long medical conversations with more speakers.

2.1. Experimental Setup

Our experiments were performed on a large corpus of about 100K ($\approx$ 15K hours) manually transcribed audio recordings of clinical conversations between physicians and patients. Apart from medical providers (physicians or nurses) and patients, the conversations may include caregivers and other support staff. In total, we model 4 speaker roles, in contrast to [4], where we restricted the experiments to 2 speaker roles.

The models are evaluated on whole conversations which are segmented automatically using a neural network based speech detector. Additionally, to understand the impact of model performance on different utterances, we created three versions of test sets with different average segment duration, specifically, 5, 10 and 15 seconds. These segments were created using the Viterbi alignment with the reference transcripts.

Table 1: *Evaluation sets with different duration.*

Segmentation	Avg (sec)	Max(sec)	Segments
Viterbi-5	7.8	389	37981
Viterbi-10	12.8	399	23125
Viterbi-15	17.8	402	16644
Automatic	38.2	60	7759
Unsegmented	674.1	3620	440

2.2. Choice of Encoder and Units

Streaming transcription of long sequences poses a challenge to the encoders. They need to capture sufficient temporal dependencies necessary for accurate prediction without observing the whole input sequence. One additional complexity that exacerbates this challenge is a mismatch between training and testing scenarios. The training data is typically segmented so that they can be trained efficiently in large batches on accelerators. Bidirectional encoders tend to be less sensitive to this mismatch, but auto-regressive encoders have been shown to exhibit high deletion rates when processing long utterances. Several workarounds have been proposed, including resetting the LSTM state every N frames [17].

Transformers with limited, fixed size, attention have been proposed for streaming applications [18]. Instead of computing attention scores over the entire sequence, this modified version of transformer adopts an approach where the computation is limited to a fixed window and the receptive field is build up hierarchically with the number of layers [19]. They demonstrated performance gains on Librispeech [20], which has relatively shorter segments than our task.

While longer units such as morphemes [21] may be beneficial in bidirectional encoders, the lack of sufficient future (right) context in unidirectional encoders could be more debilitating than smaller units like graphemes. The *output* frame rate needs to be tied to the size of the target units. This consideration led to tailoring the output rate of graphemes and morphemes to 40ms and 80ms respectively, achieved using 2 and 3 sub-sampling or TDNN layers respectively, similar to our previous work [22].

2.3. ASR Performance

The results are summarized in table 2 for streaming and non-streaming models. The key points are: (a) Transformer encoders perform about 20% better than bidirectional LSTM with TDNN layers, regardless of units or sequence lengths; (b) For the non-streaming transformer encoder, the performance does not depend on the choice of units; (c) For the full context LSTM/TDNN encoder, the performance is about 10% better when using larger morpheme units; (d) The loss of right context leads to 10% to 20% degradation in performance, with higher degradation for the larger morphemes; and (e) When using left context only, the shorter grapheme units outperform the morpheme units by about 10% in particular for longer sequences, and consistently across both encoders.

Table 2: *WER comparison of encoders and units with and without streaming constraint. Encoder Types: L=LSTM/TDNN, T=Transformer, Units: G=Graphemes, M=Morphemes, Auto: deployment scenario.*

Enc.	Units	Vit-5	Vit-10	Vit-15	Auto	UnSeg
non-streaming						
L	G	23.3	22.0	21.4	20.9	20.0
L	M	22.2	20.1	19.5	19.1	18.4
T	G	17.4	16.8	16.6	16.5	16.1
T	M	17.8	17.0	16.8	16.5	16.2
streaming						
L	G	25.5	24.7	24.4	24.8	24.4
L	M	24.1	23.2	23.3	25.1	25.5
T	G	20.7	20.0	19.7	20.3	19.6
T	M	19.8	19.0	19.5	21.9	21.2

2.4. Rich Transcription Performance

The performance of speaker segmentation and role predictions are measured in *Word Diarization Error Rate (WDER)*, the percentage of words in the transcript decorated with the right speaker tag, akin to metrics in NIST evaluations [23]. The punctuation performance is measured using *Slot Error Rate (SER)*, as defined in previous work [24, 25]. To measure capitalization performance, a special token is used to indicate an upper-case word, and we measure the slot error rate for that token.

Table 3: *Comparison of rich transcription performance under three conditions: non-streaming (NS), streaming (S) and streaming with encoder state saving (S-ESS).*

Context	WDER	WER	'.'	','	'?'	Cap
NS	3.2	16.1	35.1	48.5	36.2	25.0
S	9.7	20.3	42.7	52.0	42.8	29.5
S-ESS	5.6	19.6	-	-	-	-

Table 3 compares rich transcription performance under non-streaming (NS), streaming (S) and streaming with state saving across segments in the encoder (S-ESS). It is worth noting that extending the left context across speech segments is crucial in maintaining high accuracy in speaker role prediction when going from non-streaming (3.2% WDER in NS) to streaming scenario (5.6% WDER in S-ESS). Without the state saving mechanism, the performance drops substantially in the streaming case (9.7% WDER in S).

Our results show that capitalization is an easier task than predicting punctuation and among punctuation periods and question marks are easier than commas. There is a performance loss associated with the streaming scenario which cannot be recovered with state saving (not shown). Unfortunately, there is scant recent work on punctuation and capitalization. Models pre-trained on text corpora have been applied for post-processing ASR output to predict punctuation and capitalization [26]. They use F1-score which does not take into discounts deletions and insertions by a factor of 2 [24]. In the context of NIST Rich Transcription evaluation, the performance of predicting punctuation was reported at a slot error rate of 68% and 53.5% for ASR output and manual transcripts respectively [25]. Our results compares favorably and moreover our model does not require post-processing ASR output like the above mentioned previous work.

3. Confidence Models

There has been more than two decades of research on developing a robust and accurate confidence estimate of ASR outputs [27, 28]. In the early days when the acoustic and the language models were estimated separately, the dominant paradigm involved extracting acoustic and linguistic features from the ASR lattices and predicting the scores using a separate classifier [28]. In neural network models, one of the popular techniques involves modifying the posterior probability (softmax output) of the primary recognizer or classifier itself [29]. Unfortunately, the RNN-T models have very skewed posterior probabilities compared to, for example, CTC models which do not have an integrated language model [30]. Moreover, the RNN-T models are optimized using negative log-likelihood which can be decreased by increasing the value of logits for correct samples [31]. More recently, sequence-to-sequence models with attention have been proposed that have the flexibility to use various features from the ASR model [32, 33].

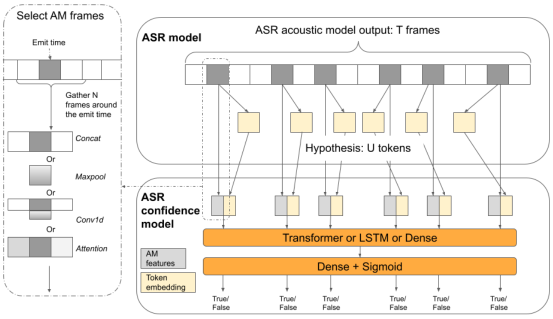

Figure 1: *Confidence model architecture.*

We propose a confidence model for RNN-T, which utilizes *acoustic and linguistic features* associated with output labels. For each hypothesized output label, we extract a latent acoustic representation from the ASR encoder outputs, stacking 3 frames of left and right contexts at the time emission of the labels. The acoustic features are augmented with the embedding of the output labels (see Figure 1) and modeled with two transformer layers and 20 unit attention window. The proposed model incurs computation cost of $O(U)$ compared to $O(TU)$ in previous work for T input frames and U output labels [33], allowing us to compute confidence over the segments even in streaming ASR applications.

While our best performing ASR systems operate at sub-word units such as graphemes and morphemes, users or downstream tasks require *confidence estimates at the world-level*. A straightforward solution involves accumulating the features at the word-level and predicting the estimates at the word-level. The downside of this approach is that the number of words are significantly lower than the number of graphemes by about a factor of 4. We investigated an alternative approach where we force the model to predict all the graphemes to be incorrect even when only one of the graphemes in the associated word is incorrect and all the others are correct. The training examples were extracted after aligning ASR hypotheses with references. At inference time, the word level confidence score is estimated by averaging the scores of the corresponding sub-word units.

The performance of confidence models is primarily measured using normalized cross-entropy (NCE) [34]. In practical applications, users or downstream applications may want to pick a confidence threshold to classify words as either correct or incorrect. Since current generation of ASR systems have substantially more correct predictions than incorrect ones, the most pertinent performance trade-off curve for picking a threshold is that with Negative Predictive Value (NPV) and True Negative Rate (TNR).

Experimental Results

The confidence models were trained on 6k held-out in addition to the training conversations and evaluated on the same ASR evaluation set. For reference, when the posterior probabilities of the (streaming morpheme) RNN-T model are used as confidence scores, the performance corresponds to an NCE of -0.40. Next, we attempted to regularize the confidence scores with the following techniques: 1) an RNN-T analog of label smoothing[1] (NCE = -0.14); 2) using temperature scaling [29] (NCE = -0.41); and 3) reducing language model overfitting by adding Gaussian noise to token embedding (NCE = -0.13), none of which helped.

Table 4: *Confidence model performances for transformer RNN-T models with grapheme (G) and morpheme (M) as target units.*

	Units	NCE	$\mathrm{AUC}_{\mathrm{NPV\ TNR}}$
non-streaming	G	0.34	0.67
	M	0.38	0.73
streaming	G	0.37	0.70
	M	0.30	0.58

The proposed confidence model substantially improves the performance, with NCE scores consistently higher than 0.3. The performance differences between the four systems were limited in terms of other metrics such as Expected Calibration Error (0.02 to 0.03), $\mathrm{AUC}_{\mathrm{ROC}}$ (0.86 to 0.89), and $\mathrm{AUC}_{\mathrm{PRC}}$ (0.95 to 0.97).

4. Extracting Medical Information

4.1. Task Description

For developing the NLP models, providers and scribes annotated useful entities and attributes in the clinical conversations [35]. The entities consisted of 64k symptoms, 100k medications, 42k conditions, 6k treatments and 38k diagnosis. The layman's description of symptoms were tagged using normalized labels. For example, ”queasy” or ”urge to vomit” were tagged with the clinical term ”nausea”. Medications were

[1] Since RNN-T loss is calculated by forward-backward algorithm, regular label smoothing [31] cannot be directly applied. Instead, we sharpened the logits in the forward pass, and used un-sharpened logits in the backward pass.

treated as an open set, and the catch all *Drug* label was applied to all direct and indirect references to drugs. This included specific and non-specific references, such as "*Tylenol*", "*the pain medication*", and "*the medication*". The attributes related to the medication entity that were annotated include *Prop:Frequency*, *Prop:Dosage*, *Prop:Mode*, and *Prop:Duration*. The conditions were categorized into – Condition:Patient, Condition:Family History, and Condition:Other.

The development and test sets were kept consistent with ASR corpus, described in Section 2.1. However, the training set is substantially smaller, with 5000 conversations, since labeling was labor-intensive.

4.2. Modeling Architecture

Linear-Chain Conditional Random Fields (CRF) are often used for named entity tagging [36]. They work well when the label set is small, but are computationally expensive if multiple ontologies are modeled jointly as in our case. A hierarchical approach decomposes the problem by using a separate span extraction and attribute tagging layer [37].

Sequence-to-Sequence with Attention models have also been used for tagging tasks [38]. The target sequence is formed as a sequence of labels, not including <O> (BIO annotation). While this model does not suffer from large label spaces, the labels are not assigned to a specific location in the input text, making the output less interpretable.

To the best of our knowledge, RNN-Ts have not been used for named entity tagging tasks yet. They offer the same benefits as other sequence-to-sequence models and, additionally, specify the location of the labels in the input. The model scales easily for large label sets, and the decoder can capture dependencies between them, e.g. the prediction of *drug:tylenol* can benefit from an earlier prediction of *sym:pain* in the sequence. Unlike the conventional tagging scheme, the target sequence contains only the begin and the end symbols of each type of tag and the location is determined by emission time steps.

4.3. Fixed Alignment Training of RNN-T

Similar to other sequence-to-sequence models, RNN-Ts learn alignments, specifically by maximizing the marginal $P(Y|X)$, computed by summing over possible alignments, $P(Y|X) = \sum_A P(Y|X, A)$. The training effectively ignores the human annotated alignment information about the location of tags in the input, making the training task harder than it needs to be.

We propose a modification that leverages the human annotation better. We encode the position information as a sequence of *blank* and *label* steps that forms a valid path in the RNN-T alignment trellis and modify the objective function such that only this specific path is used, i.e. we maximize directly $P(Y|X, A_h)$, where A_h is the alignment from the human annotations. The model does not attempt to learn alignments anymore, instead they are given. It still models the joint dependencies between inputs and outputs and the inference is performed with Viterbi beam search like regular RNN-Ts.

4.4. Experimental Results

The main results are summarized in table 5, where we compare a standard RNN-T with the fixed alignment model. The encoder is a transformer similar to the one used for ASR, and is pre-trained with a masked LM loss [39, 40] using all the 100k conversations. Note, our tagging model is the same as our ASR model, except for the fixed alignment loss. We trained the models with the sequence length of about 50 words for regular RNN-T. Training regular RNN-T models with longer sequences leads to a substantial performance drop as the alignment learning problem becomes exacerbated, while the fixed alignment RNN-T becomes robust for long sequences in inference. Overall, the fixed alignment model gives improvements over the baseline RNN-T especially for the attributes, conditions, diagnostics and treatment.

Table 5: *Named Entity Tagging Results (F-Measure). RNN-T-FA refers to the Fixed Alignment model.*

Ontology	RNN-T	RNN-T-FA	+ Self-Train
Overall	0.72	0.75	0.76
Symptoms	0.74	0.74	0.76
Medication	0.90	0.90	0.90
Attributes	0.58	0.66	0.67
Condition	0.72	0.74	0.75
Diagnostics	0.69	0.75	0.76
Treatment	0.56	0.61	0.59
Meds/Properties	0.69	0.69	0.69

In a second set of experiments, we used the 95k unlabeled conversations for self-training, as reported in the last column of table 5, we see a small improvement in symptoms, attributes, conditions and diagnostics.

4.5. Error Analysis

In previous work, we analyzed common errors observed in extracting clinical information [35]. About 30% of the residual errors were related to bad references and overall, these observations suggest that the automated metrics maybe underestimating the performance of the models. We notice similar instances in this model too. The model correctly tags the term "grass allergies", but the references misses this tag.

we know that you have SYM:IMMUNO:ALLERGIES grass allergies END:SYM because we did our DIAGS:LABS testing END:DIAGS

Another common error relates to recognizing only a portion of the relevant text. For example, the model only highlights *water* while the reference highlights the complete phrase *eyes would just water*

because I used to have this burning and my eyes would just SYM:EYES:WATEREYE water END:SYM uh-hum

5. Conclusion

In this paper, we have described three main components for extracting clinical information from medical conversations. Through empirical evaluation on real-world test set, we show that an RNN-T with a transformer encoder and grapheme units can provide better performance in a streaming (WER≈20%) scenario than LSTMs encoders or larger morpheme units. Our confidence model performs at 0.37 NCE which is comparable to other models in recent work on relatively simpler task with shorter utterances [33]. We demonstrate how a novel and relatively simpler RNN-T model with a fixed-alignment loss can improve performance over previous work that utilizes a tailored hierarchical model architecture [37].

6. Acknowledgements

We are grateful for help and support from Hasim Sak, Anshuman Tripathi, Han Lu, Qian Zhang, Nan Du, Alex Greve, Ken Su, Roberto Santana, Greg Corrado, Ryan He, and David Qiu.

7. References

[1] S. Enarvi, M. Amoia, M. Del-Agua Teba, B. Delaney, F. Diehl, S. Hahn, K. Harris, L. McGrath, Y. Pan, J. Pinto, L. Rubini, M. Ruiz, G. Singh, F. Stemmer, W. Sun, P. Vozila, T. Lin, and R. Ramamurthy, "Generating medical reports from patient-doctor conversations using sequence-to-sequence models," in *Proc. Workshop on NLP for Medical Conversations*, 2020.

[2] N. Wang, Y. Song, and F. Xia, "Studying challenges in medical conversation with structured annotation," in *Proc. Workshop on NLP for Medical Conversations*, 2020.

[3] A. Mani, S. Palaskar, and S. Konam, "Towards understanding ASR error correction for medical conversations," in *Proc. Workshop on NLP for Medical Conversations*, 2020.

[4] L. El Shafey, H. Soltau, and I. Shafran, "Joint speech recognition and speaker diarization via sequence transduction," in *Proc. Interspeech*, 2019.

[5] J. G. Fiscus, J. Ajot, M. Michel, and J. S. Garofolo, "The rich transcription 2006 spring meeting recognition evaluation," in *Proc. International Workshop on Machine Learning for Multimodal Interaction (MLMI)*, 2006, pp. 309–322.

[6] P. Woodland and D. Povey, "Large scale discriminative training of hidden markov models for speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 25–47, 2002.

[7] D. Povey, M. J. F. Gales, D. Y. Kim, and P. C. Woodland, "MMI-MAP and MPE-MAP for acoustic model adaptation," in *Proc. Interspeech*, 2003.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, vol. 30, 2017.

[10] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang, C. Fuegen, G. Zweig, and M. L. Seltzer, "Transformer-based acoustic modeling for hybrid speech recognition," in *Proc. ICASSP*, 2020.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.

[12] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proc. Interspeech*, 2017.

[13] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[14] C.-C. Chiu, A. Tripathi, K. Chou, C. Co, N. Jaitly, D. Jaunzeikare, A. Kannan, P. Nguyen, H. Sak, A. Sankar, J. Tansuwan, N. Wan, Y. Wu, and X. Zhang, "Speech recognition for medical conversations," in *Proc. Interspeech*, 2018, pp. 2972–2976.

[15] A. Graves, "Sequence transduction with recurrent neural networks," *CoRR*, 2012.

[16] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *Proc. ASRU*, 2017.

[17] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," in *Proc. Interspeech*. ISCA, 2014.

[18] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. ICASSP*, 2020, pp. 7829–7833.

[19] J. Pan, J. Shapiro, J. Wohlwend, K. J. Han, T. Lei, and T. Ma, "ASAPP-ASR: multistream CNN and self-attentive SRU for SOTA speech recognition," in *Proc. Interspeech*, 2020, pp. 16–20.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[21] P. Smit, S. Virpioja, S.-A. Grönroos, and M. Kurimo, "Morfessor 2.0: Toolkit for statistical morphological segmentation," in *Proc. EACL*, 2014, pp. 21–24.

[22] H. Soltau, H. Liao, and H. Sak, "Reducing the computational complexity for whole word models," in *Proc. ASRU*, 2017.

[23] Anonymous, "The Rich Transcription Fall 2003 (RT-03F) Evaluation Plan," NIST, Tech. Rep., 2003.

[24] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *Proc. DARPA Broadcast News Workshop*, 1999, pp. 249–252.

[25] J. Kolar and L. Lamel, "Development and Evaluation of Automatic Punctuation for French and English Speech-to-Text," in *Proc. Interspeech*, 2012.

[26] M. Sunkara, S. Ronanki, K. Dixit, S. Bodapati, and K. Kirchhoff, "Robust prediction of punctuation and truecasing for medical ASR," in *Proc. Workshop on NLP for Medical Conversations*. Association for Computational Linguistics, 2020.

[27] M. Siu and H. Gish, "Evaluation of word confidence for speech recognition systems," *Computer Speech & Language*, vol. 13, no. 4, pp. 299–319, 1999.

[28] T. Kemp and T. Schaaf, "Estimating confidence using word lattices," in *Proc. EuroSpeech*, 1997, pp. 827–830.

[29] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. ICML*, vol. 70, 2017, pp. 1321–1330.

[30] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, vol. 148, 2006, pp. 369–376.

[31] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Proc. NeurIPS*, vol. 32, 2019.

[32] Q. Li, D. Qiu, Y. Zhang, B. Li, Y. He, P. C. Woodland, L. Cao, and T. Strohman, "Confidence estimation for attention-based sequence-to-sequence models for speech recognition," in *Proc. ICASSP*, 2021.

[33] D. Qiu, Q. Li, Y. He, Y. Zhang, B. Li, L. Cao, R. Prabhavalkar, D. Bhatia, W. Li, K. Hu *et al.*, "Learning word-level confidence for subword end-to-end ASR," *arXiv preprint arXiv:2103.06716*, 2021.

[34] Anonymous, "A tutorial introduction to the ideas behind normalized cross-entropy and the information-theoretic idea of entropy," NIST, Tech. Rep., 2017.

[35] I. Shafran, N. Du, L. Tran, A. Perry, L. Keyes, M. Knichel, A. Domin, L. Huang, Y.-h. Chen, G. Li, M. Wang, L. El Shafey, H. Soltau, and J. S. Paul, "The medical scribe: Corpus development and model performance analyses," in *Proc. LREC*, May 2020.

[36] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001, p. 282–289.

[37] N. Du, K. Chen, A. Kannan, L. Tran, Y. Chen, and I. Shafran, "Extracting symptoms and their status from clinical conversations," in *Proc. ACL*, 2019, pp. 915–925.

[38] A. Kannan, K. Chen, D. Jaunzeikare, and A. Rajkomar, "Semi-supervised learning for information extraction from dialogue," in *Proc. Interspeech*, 2018, pp. 2077–2081.

[39] W. L. Taylor, ""cloze procedure": A new tool for measuring readability," *Journalism & Mass Communication Quarterly*, vol. 30, no. 4, pp. 415–433, 1953.

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL/HLT*, Jun. 2019.