# A Sequence-to-sequence Based Error Correction Model for Medical Automatic Speech Recognition

Yu Jiang
*Computer Science and Engineering*
*University of Notre Dame*
Notre Dame, IN, USA
yjiang7@nd.edu

Christian Poellabauer
*Computing and Information Sciences*
*Florida International University*
Miami, FL, USA
cpoellab@fiu.edu

2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-0126-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIBM52615.2021.9669554

***Abstract*—The use of Automatic Speech Recognition (ASR) systems in medical applications is receiving rapidly growing interest due to their ability to reduce distractions and the cognitive workload of physicians, particularly during critical medical procedures. However, state-of-the-art ASR systems still experience recognition errors, especially in noisy environments where speakers rely on medical-domain terminologies. This paper proposes a customized language model and a neural network based sequence-to-sequence (seq2seq) error correction module for medical ASR systems to provide domain adaptation and more reliable transcription results. Specifically, t he e rror correction module learns the error patterns in noisy scenarios and is able to correct such errors during inference. Our experiments show that the proposed method can reduce the sentence error rate (SER) by up to 81% for formatted input and up to 31% SER for unformatted input in noisy environments.**

***Index Terms*—medical ASR system, error correction, sequence-to-sequence model, attention mechanism**

## I. Introduction

Automatic Speech Recognition (ASR) systems often significantly i ncrease t he e fficiency an d co nvenience of human-computer interactions for a variety of applications and domains. In recent years, the medical and healthcare field is one such domain that has received increasing attention from the ASR community. For example, in hospitals, manual interactions with computing systems (e.g., recording physician notes, retrieving patient data, and searching for medical information) can be distracting, tedious, and time-consuming. A speech-based system could even be used during surgeries or emergency interventions, where such a system could also more quickly alert physicians of recommended steps or help prevent deviations from usual treatment protocols and workflows. However, there are several challenges that need to be addressed before using an ASR in the medical field will be feasible. First, the system has to work reliably in environments with different levels and types of noise, such as multiple speakers, sounds generated by medical equipment, etc. Second, it is difficult to identify a universal dataset to be used for training and evaluating medical speech recognition tasks. And third, medical terminology can be much more complex than everyday expressions, e.g., medical terms may be longer than most other dictionary words, they are often combined in unusual ways and more difficult to pronounce, while also often sharing very similar pronunciations across different words (such as names of procedures, diseases, medications, etc.). Therefore, a more advanced approach to speech recognition is needed. Since this problem is so new, there have only been a few prior efforts to investigate the design of an ASR for medical purposes [1]. One approach to designing such a system is to collect medical speech data and and build a speech corpus that can be used to train a system from scratch. Edwards et al. [2] present a speech recognition system trained with 270 hours of medical speech data and 30 million tokens of text from clinical episodes, resulting in a word error rate (WER) that is below 16% in realistic clinical cases. Chiu et al. [3] trained two models, a Connectionist Temporal Classification (CTC) phoneme based model and a Listen Attend and Spell (LAS) grapheme based model, with 14,000 hours of medical conversations, yielding WERs of 20.1% and 18.3%, respectively. Another option is to use an existing ASR system and adapt it to the medical domain. Liu et al. [4] evaluate two well-known ASR systems, Nuance Dragon and SRI Decipher, on spoken clinical questions, and adapt the SRI system to the medical domain using a language model, achieving an WER of 26.7%. Salloum et al. [5] propose a method called "crowdsourced transcription process" to continuously refine ASR language models. Mani et al. [6] perform medical domain adaptation of Google ASR and ASPIRE via machine translation, which is achieved by learning a mapping from out-of-domain errors to in-domain medical terms, yielding a WER of 7%.

In this paper, we use and adapt the ASPIRE model [7] to the medical domain to reduce the need for data collection and ASR re-training. However, to increase the performance, we propose to use a post correction module that corrects ASR outputs according to the domain the ASR is used for. The main contributions of this paper are: (1) the use of an open-source model (ASPIRE) instead of commercial models, which provides us more control over the model, (2) a machine learning based post correction module that addresses transcription errors, and (3) a model design and evaluation for speech mixed with different levels of real-world background noise, making sure that the proposed approach will be able to work well in real-world environments.

## II. Related Work

In this section, we provide an overview of ASR systems and sequence-to-sequence (seq2seq) models, while also introducing basic terminologies and concepts that are used in the proposed approach.

An ASR system integrates multiple knowledge sources (acoustic model, language model, context model, etc.) together [8], where the decoding network usually consists of the following models: a Hidden Markov Model (HMM) $H$ that outputs the acoustic probability of a phoneme in the observation of speech frames, namely the Acoustic Model (AM), a Context Model $C$ that converts triphones (context-dependent) to monophones (context-independent), a Lexicon Model $L$ that maps the phone sequences to words, and a Grammar Model $G$, namely the Language Model (LM), that encodes the probabilities of specific sentences. The commonly-used approach for knowledge source integration is mainly based on the notion of Weighted Finite State Transducers (WFSTs) [9]. With the help of WFST, each knowledge source is represented by a unified mathematical expression. Then all these WFSTs can be integrated through some transformations including composition, determinization, and minimization, which help obtain the final result and apply optimizations at the same time.

The seq2seq model has already been used widely for machine translation tasks [10]–[12], where it has been shown as an effective model. The advantage of seq2seq is that it can arbitrarily map one sequence to another sequence, and the mapping can be customized by defining attention mechanisms. Basically, seq2seq has two components: one is the Encoder, which extracts the features of the input sequence; the other is the Decoder, which predicts the output sequence based on input information. The Encoder and Decoder are usually implemented using a Recurrent Neural Network (RNN), because RNNs are capable of learning context information. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are commonly-used RNN architectures, designed for solving the problem of vanishing gradients. In addition to machine translation tasks, the seq2seq based model has also been applied to ASR error correction tasks. Wang et al. [13] utilize an augmented Transformer for entity retrieval, and Weng et al. [14] propose a word confusion network (WCN) model with multi-heads self attention for error correction and language understanding.

## III. Medical ASR System

Our medical ASR system intends to achieve two main goals. One is *domain adaptation*, i.e., being able to recognize medical terms that are uncommon in daily language. The other one is *system robustness*, i.e., maintaining satisfactory transcription accuracy in noisy environments. The proposed workflow is shown in Figure 1. First, the input is transcribed using a decoding network, which is built based on the ASPIRE model. The domain adaptation to the medical field is obtained through re-training the LM. Then the results are further optimized using a pre-trained seq2seq model, which has previously learned the error patterns.

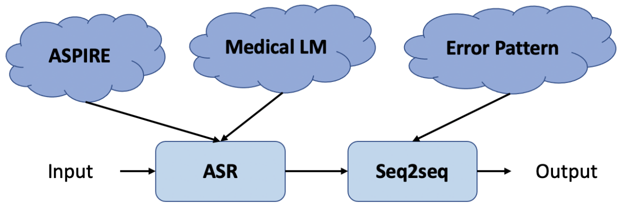

Fig. 1. Workflow of the proposed method.

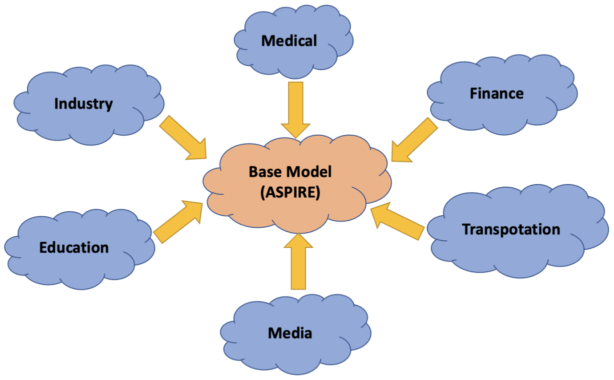

Fig. 2. Adaptation to multiple different domains.

### A. Domain Adaptation

A common way for domain adaptation is to train the AM and the LM for the medical field from scratch. However, training a good AM typically requires a large speech dataset. It is not easy to collect sufficient medical data in real applications; therefore, we propose to build the system based on one general model – the ASPIRE model. We then directly apply the AM of ASPIRE and only re-train the LM for the medical domain.

The medical ASR system decodes physicians' medical commands and queries, where the transcriptions can be utilized by various higher layer applications, e.g., to record the medical procedures or medications given to a patient, to provide guidance and support to a physician, and to prevent medical errors. There are two types of ASR inputs: one is *formatted input*, i.e., the input follows specific patterns, the other type is *unformatted input*, i.e., there are many varieties of sentence structures. An example of formatted input is "amlodipine 10 mg", which follows the pattern of NAME + DOSAGE + UNIT. An example of unformatted input is "Is there history of use of seretide", which is more similar to common real-world user expressions.

We encounter a conflict when training the medical LM, i.e., on one hand, we hope that the medical LM could be a universal one, because once the decoding network is deployed, it would be costly to modify it. On the other hand, the medical LM

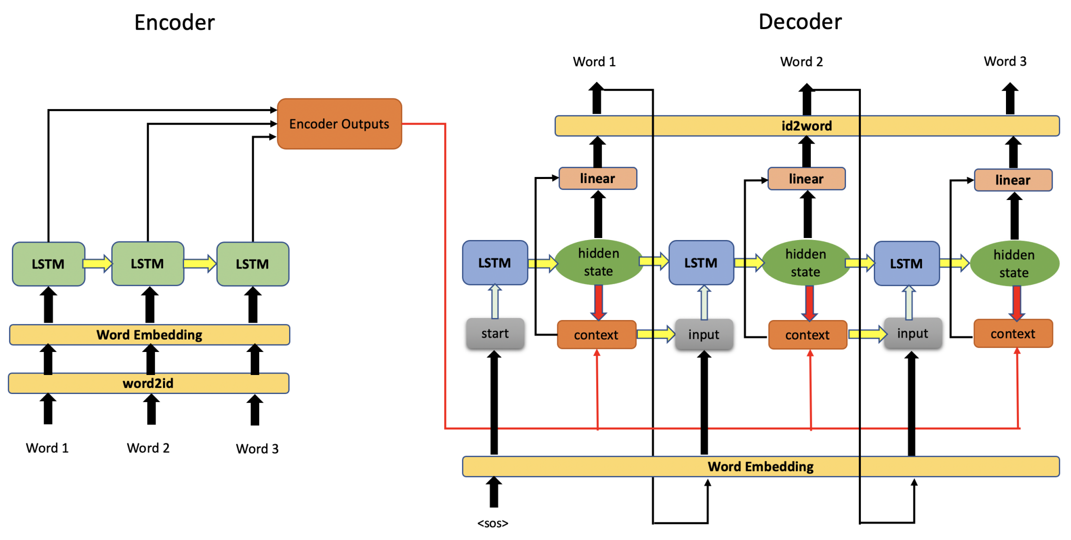

Fig. 3. Overview of the seq2seq model with attention mechanism.

would be imbalanced if trained from the imbalanced medical corpus itself. An imbalanced LM means that it can distinguish high-frequency input from low-frequency one. For example, if "aspirin 15 mg" is shown more often than "aspirin 50 mg" in the corpus, then the former one will have higher probability. However, this kind of imbalance has strong dependence on the corpus and it can easily change when there is a different scenario. It is reasonable to assume that "Command 1" will have a higher probability than "Command 2" in one circumstance, while a higher probability of "Command 2" will may make sense under other circumstances. To eliminate this kind of imbalance and to maintain equal probabilities for all inputs, we apply a different training strategy rather than directly training the LM from medical corpus.

The solution is to store the input patterns rather than the content. For formatted input, storing pattern means that all the inputs following the same pattern will have the same probability, no matter what entity exactly it is, e.g., "aspirin 50 mg" and "amlodipine 10 g" will have the same probability, because they follow the same pattern of NAME + DOSAGE + UNIT. For unformatted input, the situation is more complicated, because we cannot list all the combinations of different entities as with the formatted input. One possible approach is to find similar words in the ASPIRE LM for each new medical word and define the n-gram parameter values of new words [15]. This approach is beyond the scope of the paper; instead, in this paper, propose to train the LM directly for unformatted input. This way, when we need to perform adaptations for different circumstances, it is not necessary to modify the decoding network. We can maintain different LMs, which are independent of the decoding network, to re-score the outputs. The advantage is that this kind of structure can then easily be updated.

This basic framework can be applied to multiple domain adaptations, which is shown in Figure 2. We can simply build one universal ASR system for recognition tasks in different domains, e.g., medical, educational, industry, etc., instead of designing different ones for each separate domain. This is achieved by merging the LMs of different domains and keeping the rest of the components of ASPIRE unchanged. It is also possible to highlight some "hot" domains or eliminate the domain imbalance due to unequal dataset sizes by assigning different weights to different domains. This way, we can save a lot of overheads, i.e., we do not need to collect large amounts of speech data for model re-training and maintain multiple ASR systems for different domains. This will be increasingly important for ASR systems on resource-constrained edge or end devices and when domain changes for the same system may be common.

## B. *Seq2seq Based Post Correction Module*

A seq2seq model can be very effective in decreasing decoding error rates. There exists an error pattern when transcribing noisy speech; specifically, particular words only have a limited number of incorrect transcriptions. When a word would be transcribed as "A" or as "B" depends on the context. The seq2seq model is capable of learning this pattern, acquiring the knowledge of frequent mappings, and making the prediction contextually. Therefore, it is possible to recover the medical input from the erroneous decoding results.

The model structure used in this paper is shown in Figure 3. The encoder consists of one word2id layer, one word embedding layer with embedding size of 512, and one LSTM layer with hidden layer size of 512. The decoder also consists of one word embedding layer and one LSTM layer of the same size as the encoder. Further, it includes one id2word layer and one linear layer. The input sentence, the ASR decoding result, is first broken down into a sequence of words, and then each word is translated into a word embedding. Then the LSTM layer extracts the features of the input and provides them to the decoder. Finally, the LSTM layer of the decoder makes the prediction sequentially based on the input information and prediction history. The loss function is a cross entropy loss and the outputs are the probabilities of every possible word. A linear layer is required to unify the dimensions of the LSTM and the output.

Additionally, an attention mechanism is introduced to our post correction model. The standard seq2seq model only takes the last hidden state as the input of the decoder, so the impact of past input becomes weaker and weaker as the sequence length increases. An attention mechanism directly addresses this issue by retaining and utilizing all the hidden states of the encoder during the decoding process. It assigns different weights to encoder outputs and generates unique context vectors at different time steps of the decoder. With the help of the attention mechanism, the correction can be more accurate by leveraging more comprehensive contextual information.

There are two major types of attention: Bahdanau Attention [16] and Luong Attention [17]. The main difference is at which position the attention is being introduced. Further, the alignment scores are calculated in different ways. There are three types of calculation for the Luong Attention, while the Bahdanau Attention only has one single type as follows.
Bahdanau Attention:

$$\begin{aligned} score = W_{combined} \cdot \tanh(&W_{decoder} \cdot H_{decoder} \\ &+ W_{encoder} \cdot H_{encoder}) \end{aligned} \tag{1}$$

Dot Attention:

$$score = H_{encoder} \cdot H_{decoder} \tag{2}$$

General Attention:

$$score = W \cdot H_{encoder} \cdot H_{decoder} \tag{3}$$

Concat Attention:

$$score = W \cdot \tanh(W_{combined} \cdot (H_{encoder} + H_{decoder})) \tag{4}$$

where $H_{encoder}$ and $H_{decoder}$ are encoder output and decoder output, respectively, and $W$, $W_{encoder}$, $W_{decoder}$, and $W_{combined}$ represent different weight matrices.

Here we apply the Luong Attention with a multilayer perceptron (MLP) and calculate alignment scores using the Dot Attention function.

## IV. Experimentation

Figure 4 shows the steps of the experimentations performed in this work. It also presents the knowledge sources and tools that are used in each stage. We have two independent types of medical input: formatted and unformatted. The dataset is generated as follows. We generate the text of the medical input, convert the text to speech, and synthesize the clean speech and the noise. Then we evaluate two approaches; one is to directly decode using our medical ASR system, and the other is to continually optimize the transcription with the help of the seq2seq based post correction model.

### *A. Dataset Creation*

First we demonstrate the creation of the formatted dataset, which contains drug ordering commands. The legal drug names are obtained from the U.S. National Library of Medicine (NIH). The original list has 3051 drug names. We format the list by removing the names that contain numbers or symbols, and then obtain 681 names. For each drug, we randomly generate its valid dosage in order to simulate the actual correct scenario, because in practical scenarios only some specific dosages would be ordered by a physician. Note that each drug does not necessarily have only one valid dosage; different drugs have different numbers of valid dosages. After we obtain all these candidate names and their corresponding dosages, we generate 30,000+ medical commands by combining the entities – name, dosage, and unit. All the commands in this dataset follow the same pattern of NAME + DOSAGE + UNIT.

The unformatted dataset is created through directly making use of the emrQA [18] datasets, which are domain-specific large-scale question answering (QA) datasets generated by re-purposing existing annotations for other NLP tasks. They consist of 400,000+ question-answer pairs, but we only use the medical portions. In the beginning we have over 1,300,000 medical questions, and 138,456 questions are left after we perform data cleaning, i.e., deleting the questions that contain numbers or symbols.

After we acquire the text of the medical input, it is converted to speech with the help of Google Text-To-Speech (TTS). To make the situation more practical, the clean speech of the medical input is mixed with real-world noises such as air-conditioner sounds, babbling, copy machine sounds, and vacuum cleaners, which are all included in the Microsoft Scalable Noisy Speech Dataset (MS-SNSD) [19]. Each dataset is divided into 3 different sets for both the formatted and unformatted data. For the formatted data, we have 30,000 samples for training, 3,000 samples for validation, and 3,000 samples for testing. For the unformatted data, we have 110,766 samples for training, 13,845 samples for validation, and 13,845 samples for testing.

### *B. ASR System Implementation*

The ASR system is implemented using Kaldi. We make use of the ASPIRE model and adapt it to the medical domain by replacing the original LM with the medical one, which is trained from the medical corpus (explained in Section III-A). In the LM training, we use the n-gram model and apply the algorithm of Ney's absolute discounting with a smoothing parameter of 0.5. All other components for constructing the ASR decoding networks are unchanged, i.e., we directly use the pre-trained ASPIRE model.

The decoding utility in Kaldi used in this work is "online2-wav-nnet3-latgen-faster", which provides fast and accurate decoding. We turn off online decoding to maximize accuracy. The parameter of "max-active" is 7000, which controls the maximum number of states that can be active at one time; "beam" is 15, which controls the process of beam pruning; and

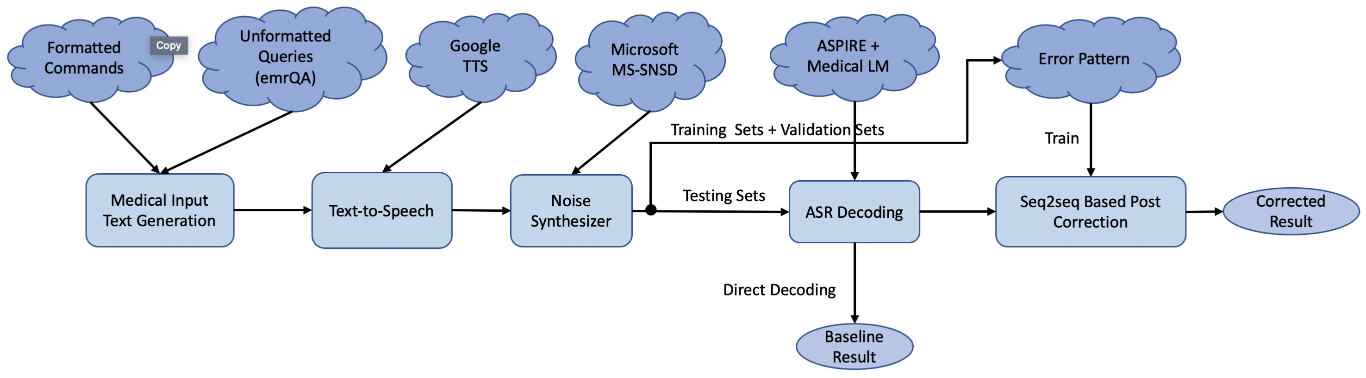

Fig. 4. Experimental setup and steps.

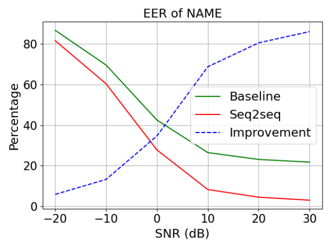

Fig. 5. EER of name.

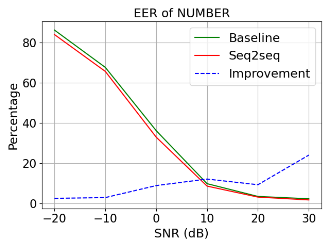

Fig. 6. EER of number.

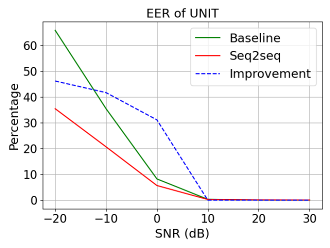

Fig. 7. EER of unit.

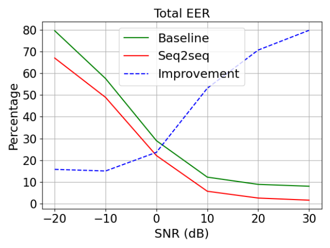

Fig. 8. Total EER.

TABLE I
EVALUATION OF CORRECTION PERFORMANCE OF EACH ENTITY AT DIFFERENT SNR LEVELS (RR REPRESENTS RIGHT TO RIGHT AND RW REPRESENTS RIGHT TO WRONG)

| **SNR** | **Name Entity** | | | | | | **Number Entity** | | | | | | **Unit Entity** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **(dB)** | ***RR*** | ***RW*** | ***WR*** | ***WW*** | ***WCR*** | ***RCR*** | ***RR*** | ***RW*** | ***WR*** | ***WW*** | ***WCR*** | ***RCR*** | ***RR*** | ***RW*** | ***WR*** | ***WW*** | ***WCR*** | ***RCR*** |
| -20 | 392 | 7 | 160 | 2441 | 0.018 | 0.062 | 390 | 26 | 93 | 2491 | 0.062 | 0.036 | 1011 | 15 | 926 | 1048 | 0.015 | 0.469 |
| -10 | 893 | 19 | 296 | 1792 | 0.021 | 0.142 | 877 | 95 | 155 | 1873 | 0.098 | 0.076 | 1912 | 25 | 468 | 595 | 0.013 | 0.44 |
| 0 | 1695 | 30 | 472 | 803 | 0.017 | 0.37 | 1801 | 113 | 210 | 876 | 0.059 | 0.193 | 2735 | 18 | 95 | 152 | 0.007 | 0.385 |
| 10 | 2190 | 16 | 562 | 232 | 0.007 | 0.708 | 2629 | 75 | 111 | 185 | 0.028 | 0.375 | 2984 | 6 | 8 | 2 | 0.002 | 0.8 |
| 20 | 2299 | 8 | 566 | 127 | 0.003 | 0.817 | 2832 | 62 | 72 | 34 | 0.021 | 0.679 | 2996 | 3 | 0 | 1 | 0.001 | 0.0 |
| 30 | 2336 | 11 | 573 | 80 | 0.005 | 0.877 | 2883 | 46 | 63 | 8 | 0.016 | 0.887 | 2998 | 2 | 0 | 0 | 0.001 | 0.0 |

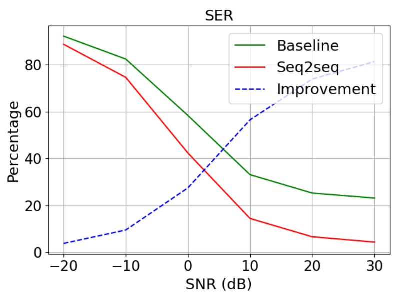

Fig. 9. SER of formatted dataset with various SNRs.

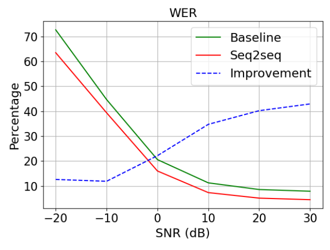

Fig. 10. WER of unformatted dataset with various SNRs.

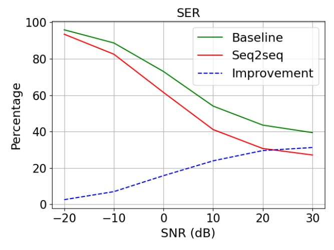

Fig. 11. SER of unformatted dataset with various SNRs.

"acoustic-scale" is 1, which determines the weight of acoustic probabilities.

### *C. Training the Seq2seq Model*

First we decode all the speech samples of the two datasets, i.e., formatted and unformatted. Then we train the neural network based on the training sets: the direct decoding result from the ASR system is the neural net input and the correct medical text is the neural net output, i.e., the input is an erroneous transcription and the output is the label. We adjust the hyper-parameters to minimize the loss on the validation set, so the network can learn the error patterns better and make corrections more accurately. The network settings are: the batch size is 32, the epoch is 50, and the optimizer is Adam [20] with a learning rate of 1e-4. Additionally, we apply the mechanisms of teacher forcing learning and gradient clipping.

### *D. Results and Analysis*

We evaluate different metrics for the two different dataset. We analyze two metrics for the formatted dataset: entity error rate (EER) and sentence error rate (SER), and two metrics for the unformatted dataset: word error rate (WER) and SER.

*1) Formatted Dataset:* According to the dataset, we know that we have 3 entities: name, dosage, and unit. Each entity is important in medical treatment, and none of the errors are negligible. Our goal is to reduce SER, but the EER is also an important metric that can help us understand which parts of the model will need to be improved further.

The experiments are carried out using different levels of SNR: -20 dB to 30 dB. We take the direct decoding as the baseline and compare it with the proposed method. The EER of each entity, total EER, and SER are shown in Figures 5-9, respectively. The improvement represents a relative value rather than an absolute one compared to the baseline. Observing the results, we find that the proposed method does achieve good improvement through the seq2seq optimization. The improvement of SER is surprising with higher SNR: the SER decreases from 23.1% to 4.3%, i.e., it decreases by more than 80% when SNR is 30 dB. The improvement is also satisfactory with medium SNRs: the SER decreases from 33.1% to 14.4%, i.e., it decreases by 56.5% when SNR is 10 dB. We also notice that the proposed method is not as effective when the speech quality is extremely low (lower than 0 dB): the SER is still above 80% after the seq2seq model. In this case the ASR system cannot provide useful information for the seq2seq model to recover the commands, i.e., the noise power is too strong.

We can also learn something from the EER of each entity. Table I shows the correction performance of each entity at different SNR levels, where RR represents "right to right", i.e., the direct ASR decoding result is right and the result is still right after seq2seq based error correction. Similarly, RW represents "right to wrong", WR represents "wrong to right" and WW represents "wrong to wrong". Wrong correction rate (WCR) and right correction rate (RCR) are calculated as (5) and (6), which reflect the correction performance in two aspects: the probability of incorrect correction and the capability of recovering from mistakes.

$$WCR = RW/(RR + RW) \tag{5}$$

$$RCR = WR/(WR + WW) \tag{6}$$

The results show that name entity contributes most to the improvement: it has higher RCR (above 70%) and lower WCR (below 10%) when SNR is rather high (higher than 0 dB). And number entity is more difficult to be corrected: the RCR is not as high and the WCR is also not as low. As for unit entity, we only have limited candidates for it, so direct decoding is enough for the recognition.

*2) Unformatted Dataset:* It is not easy to extract entities from the unformatted sentences. Therefore, we simply calculate WER rather than EER to make the problem easier.

The experiments are the same as for the formatted dataset, i.e., using different SNR levels from -20 dB to 30 dB. The baseline is direct decoding and it is compared with the proposed method. The WER and SER values are shown in Figures 10-11, respectively. The results show that the proposed method does well in improving the decoding accuracy with higher SNRs: when SNR is 30 dB, WER is decreased by 42.9% from 7.9% to 4.5%, while SER is decreased by 31.2% from 39.4% to 27.1%. It is also verified that the post correction model cannot help in processing low-quality speech: SER is above 80% when SNR is lower than -10 dB.

*3) Conclusions for Experimental Results:* Comparing the experimental results of the formatted and unformatted datasets, the effectiveness of the proposed method is shown. When SNR is higher than 0 dB, the decoding accuracy can be significantly improved with the help of the seq2seq based post correction model, which provides great support for downstream tasks such as language understanding. However, the proposed method also has limitations, i.e., it does not perform well when the speech is completely submerged in noise, i.e., SNR is lower than 0 dB (note, however, that the real-world impact of this is limited since 0 dB is the so-called hearing threshold for the human ear and speech with SNR below 0 dB is very uncommon in real application scenarios).

## V. Conclusions and Future Work

In this paper, we propose an ASR system model that can recognize medical speech more accurately. We apply a domain-specific ASR model together with a seq2seq based post correction module. We transfer the ASR model to the medical domain by re-training the LM of the ASPIRE model using a medical corpus, and we apply a seq2seq neural network with attention mechanism in the post correction module. We evaluate the performance of the model on two datasets, i.e., the formatted and unformatted datasets, which consist of speech data synthesized based on medical text. The results show that our system model reduces the SER by up to 81% compared to direct decoding for formatted input and up to 31% for unformatted input. The improvement of accuracy decreases as the SNR goes down, and the system is not as effective when the speech quality is extremely bad. Particularly for the formatted dataset, only the name entity can be better corrected, while the other entities are not corrected very well. Therefore, in our future work, we will (1) further study techniques to separately address transcription errors using entity-specific techniques, especially for number and unit entity, (2) collect medical speech data in real applications and evaluate the performance of the proposed method, and (3) introduce more context knowledge, e.g., subject, speaker, procedure, to the system so that it can acquire more intelligent inference ability.

## VI. Acknowledgement

We acknowledge Dr. Yuan Gong for giving suggestions for this project.

## References

[1] S. Latif, J. Qadir, A. Qayyum, M. Usama, and S. Younis, "Speech technology for healthcare: Opportunities, challenges, and state of the art," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 342–356, 2021.

[2] E. Edwards, W. Salloum, G. P. Finley, J. Fone, G. Cardiff, M. Miller, and D. Suendermann-Oeft, "Medical speech recognition: Reaching parity with humans," in *Speech and Computer*. Cham: Springer International Publishing, 2017, pp. 512–524.

[3] 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 *Interspeech*, 2018, pp. 2972–2976.

[4] F. Liu, G. Tur, D. Hakkani-Tur, and H. Yu, "Towards spoken clinical-question answering: Evaluating and adapting automatic speech-recognition systems for spoken clinical questions," *Journal of the American Medical Informatics Association : JAMIA*, vol. 18, pp. 625–30, 06 2011.

[5] W. Salloum, E. Edwards, S. Ghaffarzadegan, D. Suendermann-Oeft, and M. Miller, "Crowdsourced continuous improvement of medical speech recognition," in *AAAI Workshops*, 2017.

[6] A. Mani, S. Palaskar, N. V. Meripo, S. Konam, and F. Metze, "Asr error correction and domain adaptation using machine translation," in *Proc. IEEE ICASSP*, 2020, pp. 6344–6348.

[7] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, "Jhu aspire system: Robust lvcsr with tdnns, ivector adaptation and rnn-lms," in *IEEE Workshop on ASRU*, 2015, pp. 539–546.

[8] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," 2011.

[9] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlíček, Y. Qian, K. Riedhammer, K. Veselý, and N. T. Vu, "Generating exact lattices in the wfst framework," in *Proc. IEEE ICASSP*, 2012, pp. 4213–4216.

[10] Y. Wu, M. Schuster, Z. Chen *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[11] J. Crego, J. Kim, G. Klein *et al.*, "Systran's pure neural machine translation systems," *arXiv preprint arXiv:1610.05540*, 2016.

[12] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, "Deep recurrent models with fast-forward connections for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 371–383, 07 2016.

[13] H. Wang, S. Dong, Y. Liu, J. Logan, A. K. Agrawal, and Y. Liu, "Asr error correction with augmented transformer for entity retrieval," in *Interspeech*, 2020, pp. 1550–1554.

[14] Y. Weng, S. S. Miryala, C. Khatri, R. Wang, H. Zheng, P. Molino, M. Namazifar, A. Papangelis, H. Williams, F. Bell, and G. Tur, "Joint contextual modeling for asr correction and language understanding," in *Proc. IEEE ICASSP*, 2020, pp. 6349–6353.

[15] L. Orosanu and D. Jouvet, "Adding new words into a language model using parameters of known words with similar behavior," *Procedia Computer Science*, vol. 128, pp. 18–24, 2018, 1st International Conference on Natural Language and Speech Processing.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2016.

[17] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[18] A. Pampari, P. Raghavan, J. Liang, and J. Peng, "emrqa: A large corpus for question answering on electronic medical records," *arXiv preprint arXiv:1809.00732*, 2018.

[19] C. K. Reddy, E. Beyrami, J. Pool, R. Cutler, S. Srinivasan, and J. Gehrke, "A scalable noisy speech dataset and online subjective test framework," pp. 1816–1820, 2019.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.