Towards Understanding ASR Error Correction for Medical Conversations

Anirudh Mani
Abridge AI Inc.
amani@abridge.com

Shruti Palaskar
Carnegie Mellon University
spalaska@cs.cmu.edu

Sandeep Konam
Abridge AI Inc.
san@abridge.com

Abstract

Domain Adaptation for Automatic Speech Recognition (ASR) error correction via machine translation is a useful technique for improving out-of-domain outputs of pre-trained ASR systems to obtain optimal results for specific in-domain tasks. We use this technique on our dataset of Doctor-Patient conversations using two off-the-shelf ASR systems: Google ASR (commercial) and the ASPIRE model (open-source). We train a Sequence-to-Sequence Machine Translation model and evaluate it on seven specific UMLS Semantic types, including Pharmacological Substance, Sign or Symptom, and Diagnostic Procedure to name a few. Lastly, we breakdown, analyze and discuss the 7% overall improvement in word error rate in view of each Semantic type.

1 Introduction

Off-the-shelf ASR systems like Google ASR are becoming increasingly popular each day due to their ease of use, accessibility, scalability and most importantly, effectiveness. Trained on large datasets spanning different domains, these services enable accurate speech-to-text capabilities to companies and academics who might not have the option of training and maintaining a sophisticated state-of-the-art in-house ASR system. However, for all the benefits these cloud-based systems provide, there is an evident need for improving their performance when used on in-domain data such as medical conversations. Approaching ASR Error Correction as a Machine Translation task has proven to be useful for domain adaptation and resulted in improvements in word error rate and BLEU score when evaluated on Google ASR output (Mani et al., 2020).

However, it is important to analyze and understand how domain adapted speech may vary from ASR outputs. We approach this problem by using two different types of metrics - 1) overall transcription quality, and 2) domain specific medical information. For the first one, we use standard speech metric like word error rate for two different ASR system outputs, namely, Google Cloud Speech API[1] (commercial), and ASPIRE model (open-source) (Peddinti et al., 2015). For the second type of evaluation, we use the UMLS[2] ontology (O., 2004) and analyze the S2S model output for a subset of semantic types in the ontology using

Model	Transcript
Reference	you also have a *pacemaker* because you had sick sinus syndrome and it's under control
Google ASR	you also have a **taste maker** because you had sick sinus syndrome and it's under control
S2S	you also have a **pacemaker** because you had sick sinus syndrome and it's under control
Reference	like a heart disease uh *atrial fibrillation*
Google ASR	like a heart disease **asian populations**
S2S	like a heart disease **atrial fibrillation**

Table 1: Examples from Reference, Google ASR transcription and corresponding S2S model output for two medical words, *"pacemaker"* and *"atrial fibrillation"*. In this work, we investigate how adapting transcription to domain and context can help reduce such errors, especially with respect to medical words categorized under different Semantic types of the UMLS ontology.

[1] https://cloud.google.com/speech-to-text/

[2] The Unified Medical Language System is a collection of medical thesauri maintained by the US National Library of Medicine

Proceedings of the 1st Workshop on NLP for Medical Conversations, pages 7–11
Online, 10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

a variety of performance metrics to build an understanding of effect of the Sequence to Sequence transformation.

2 Related Work

While the need for ASR correction has become more and more prevalent in recent years with the successes of large-scale ASR systems, machine translation and domain adaptation for error correction are still relatively unexplored. In this paper, we build upon the work done by Mani et al. (Mani et al., 2020). However, D'Haro and Banchs (D'Haro and Banchs, 2016) first explored the use of machine translation to improve automatic transcription and they applied it to robot commands dataset and human-human recordings of tourism queries dataset. ASR error correction has also been performed based on ontology-based learning in (Anantaram et al., 2018). They investigate the use of including accent of speaker and environmental conditions on the output of pre-trained ASR systems. Their proposed approach centers around bio-inspired artificial development for ASR error correction. (Shivakumar et al., 2019) explore the use of noisy-clean phrase context modeling to improve ASR errors. They try to correct unrecoverable errors due to system pruning from acoustic, language and pronunciation models to restore longer contexts by modeling ASR as a phrase-based noisy transformation channel. Domain adaptation with off-the-shelf ASR has been tried for pure speech recognition tasks in high and low resource scenarios with various training strategies (Swietojanski and Renals, 2014, 2015; Meng et al., 2017; Sun et al., 2017; Shinohara, 2016; Dalmia et al., 2018) but the goal of these models was to build better ASR systems that are robust to domain change. Domain adaptation for ASR transcription can help improve the performance of domain-specific downstream tasks such as medication regimen extraction (Selvaraj and Konam, 2019).

3 Domain Adaptation for Error Correction

Using the reference texts and pre-trained ASR hypothesis, we have access to parallel data that is in-domain (reference text) and out-of-domain (hypothesis from ASR), both of which are transcriptions of the same speech signal. With this parallel data, we now frame the adaptation task as a translation problem.

Sequence-to-Sequence Models : Sequence-to-sequence (S2S) models (Sutskever et al., 2014) have been applied to various sequence learning tasks including speech recognition and machine translation. Attention mechanism (Bahdanau et al., 2014) is used to align the input with the output sequences in these models. The encoder is a deep stacked Long Short-Term Memory Network and the decoder is a shallower uni-directional Gated Recurrent Unit acting as a language model for decoding the input sequence into either the transcription (ASR) or the translation (MT). Attention-based S2S models do not require alignment information between the source and target data, hence useful for monotonic and non-monotonic sequence-mapping tasks. In our work, we are mapping ASR output to reference hence it is a monotonic mapping task where we use this model.

4 Experimental Setup

4.1 Dataset

We use a dataset of 3807 de-identified Doctor-Patient conversations containing 288,475 utterances split randomly into 230,781 training utterances and 28,847 for validation and test each. The total vocabulary for the machine translation task is 12,934 words in the ASR output generated using Google API and ground truth files annotated by humans in the training set. We only train word-based translation models in this study to match ASR transcriptions and ground truth with further downstream evaluations. To choose domain-specific medical words, we use a pre-defined ontology by Unified Medical Language System (UMLS) (O., 2004), giving us an exhaustive list of over 20,000 medications. We access UMLS ontology through the Quickumls package (Soldaini and Goharian, 2016), and use seven semantic types - Pharmacological Substance (PS), Sign or Symptom (SS), Diagnostic Procedure (DP), Body Part, Organ, or Organ Component (BPOOC), Disease or Syndrome (DS), Laboratory or Test Result (LTR), and Organ or Tissue Function (OTF). These are thereby referred by their acronyms in this paper. These seven semantic types were chosen to cover a spread of varied number of utterances available for each type's presence, from lowest (OTF) to the highest (PS) in our dataset.

Alignment: Since the ground truth is at utterance level, and ASR system output transcripts are

Ontology	Utts	Unique Terms
	Train, Test	Train, Test
PS	35301, 4481	1233, 532
DS	17390, 2191	859, 310
BPOOC	15312, 1944	513, 222
SS	14245, 1805	429, 181
DP	4016, 484	217, 82
LTR	3466, 407	70, 33
OTF	1866, 228	68, 26

Table 2: Breakdown of the Full Data based on REF.

at word level, specific alignment handling techniques are required to match the output of multiple ASR systems. This is achieved using utterance level timing information i.e., start and end time of an utterance, and obtaining the corresponding words in the ASR system output transcript based on word-level timing information (start and end time of each word). To make sure same utterance ID is used across all ASR outputs and the ground truth, we first process our primary ASR output transcripts from Google Cloud Speech API based on the ground truth and create random training, validation and test splits. For each ground truth utterance in these dataset splits, we also generate corresponding utterances from ASPIRE output transcripts similar to the process mentioned above. This results in two datasets corresponding to Google Cloud Speech and ASPIRE ASR models, where utterance IDs are conserved across datasets. However, this does lead to ASPIRE dataset having a lesser utterances as we process Google ASR outputs first in an effort maximize the size of our primary ASR model dataset.

Pre-trained ASR: We use the Google Cloud Speech API for Google ASR transcription and the JHU ASPIRE model (Peddinti et al., 2015) as two off-the-shelf ASR systems in this work. Google Speech API is a commercial service that charges users per minute of speech transcribed, while the ASPIRE model is an open-source ASR model. We explore the trends we observe in both–a commercial API as well as an open-source model.

5 Results and Discussions

5.1 Transcription Quality

We use WER and BLEU scores to evaluate improvement on ASR model outputs using the S2S model. A consistent gain is observed across all metrics, with an absolute improvement of 7% in WER and a 4 point absolute improvement in BLEU scores on Google ASR. While the Google ASR output can be stripped of punctuation for a better comparison, it is an extra post-processing step and breaks the direct output modeling pipeline. If necessary, ASPIRE model output and the references can be inserted with punctuation as well.

Transcript	WER (⇓)	BLEU (⇑)
Google ASR output	41.0	52.1
+ S2S Adapted	34.1	56.4
ASPIRE ASR output	35.8	54.3
+ S2S Adapted	34.5	55.8

Table 3: Results for adaptive training experiments with Google ASR and ASPIRE model. We compare absolute gains in WER and BLEU scores with un-adapted ASR output.

5.2 Qualitative Analysis

In Table 4, we compare S2S adapted outputs with Google ASR for each semantic type, broken down by Precision, Recall and F1 scores. The two outputs are also compared directly by counting utterances where S2S model made the utterance better with respect to a semantic term - it was present in the reference and S2S output but not Google ASR, and cases where S2S model made the utterance worse - semantic term was present in the reference and Google ASR but not S2S output. We refer to this metric as *semantic intersection* in this work.

As observed, the F1 scores are higher for S2S outputs for all the semantic types in the Ontology, except for one (BPOOC) where it ties. In terms of Precision and Recall too, S2S performs better for most categories. These numbers can be discussed with a couple of underlying factors - how common or rare the semantic terms are on average for each semantic type, and how many training examples has the model seen for those terms. This is important to consider as Google ASR learns on a much larger vocabulary of words spanning many different domains, where as S2S is trained on a domain specific dataset. For example, we see a large gain on Precision for DP, which can be attributed to the rarity of the terms under this category, like 'echocardiogram', 'pacemaker', etc. Its also for this reason we see only a slight improvement in Precision for PS even though it has the most number of training examples. Many of the medication names are rare, but a lot of them are pretty common

Ontology	Unique Terms	S2S adpt, ASR o/p			
	S, G, R	P	R	F1	SI
PS	282, 393, 532	**0.86** , 0.85	**0.61** , 0.55	**0.72** , 0.67	**0.10**, 0.02
DS	210, 302, 310	0.75 , 0.75	0.68 , 0.68	**0.76** , 0.75	**0.03**, 0.02
BPOOC	173, 235, 222	**0.82** , 0.81	0.70 , 0.70	0.75 , 0.75	0.02, 0.02
SS	144, 169, 181	0.87 , **0.88**	**0.74** , 0.72	**0.8** , 0.79	**0.03**, 0.01
DP	54, 73, 82	**0.89** , 0.75	0.65 , **0.70**	**0.75** , 0.72	0.02, **0.07**
LTR	26, 26, 33	0.77 , **0.85**	**0.67** , 0.61	**0.72** , 0.71	**0.07**, 0.01
OTF	26, 32, 26	**0.79** , 0.74	**0.79** , 0.77	**0.79** , 0.75	**0.04**, 0.02

Table 4: Medical WER results per Ontology for adaptive training experiments on Test data. We use Precision, Recall, F1 and Semantic Intersection (as defined in 5.2) metrics for comparing S2S model output to Google ASR.

nowadays even though they are domain specific, like ‘aspirin’. Moreover, this is also supported by the numbers observed for BPOOC, where terms like ‘legs’, ‘heart’ and ‘lungs’ are the top 3 most frequently occurring words.

The number of unique terms for the S2S output are lower in comparison to Google ASR and reference as observed in Table 4. This might indicate that the S2S model is incorrectly modifying some Google ASR output medical terms which may not have as many examples in the Training set. However, our *semantic intersection* metric indicates that we get an overall improvement in all categories, except for DP. We hypothesize this to be largely due to a combination of how rare the words are, and the overall number of training examples for DP being low. When we calculate *semantic intersection* on the Full set, we get almost equal results for S2S and Google ASR outputs, 0.5 and 0.6 respectively. When we look at our top 5 and bottom 5 least frequent terms for each semantic types, almost all the terms overlap between S2S, Google ASR and reference, even though the number of unique terms might be less for S2S. Overall, it is evident from analyzing the results that as the number of occurrences increases for each medical term, the performance of the S2S model in identifying errors and correcting them increases rapidly, as shown in Table 2 and Table 4.

In a production environment, the S2S model may be confidently used for correcting ASR errors for top K most frequently occurring medical terms, where the value of K must be decided based on the dataset available for training. Future extension of this work will also be looking into the class imbalance problem for a more robust performance on different semantic types.

6 Conclusion

We present an analysis of how ASR Error Correction using Machine Translation impacts the different semantic types of the UMLS ontology for a medical conversation. We run the S2S model on a dataset of Doctor-Patient conversations as a post-processing step to optimize the Google off-the-shelf ASR system. We use different input representations and compare the performance of our S2S model using WER and BLEU scores on Google ASR and ASPIRE outputs. We deep dive into how our adaptation model affect medical WER for each semantic type, and breakdown the results using Precision, Recall, F1 and Semantic Intersection numbers between S2S and Google ASR. We establish the robustness of S2S model performance for more frequently occurring medical terms. In the future, we want to explore other representations like phonemes which might capture ASR errors better, and address the class imabalance problem for rarer medical terms in different semantic types.

Acknowledgments

We thank the University of Pittsburgh Medical Center (UPMC) and Abridge AI Inc. for providing access to de-identified data of Doctor-Patient conversations used in this work.

References

C Anantaram, Amit Sangroya, Mrinal Rawat, and Aishwarya Chhabra. 2018. Repairing asr output by artificial development and ontology based learning. In *IJCAI*, pages 5799–5801.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Siddharth Dalmia, Xinjian Li, Florian Metze, and Alan W Black. 2018. Domain robust feature extraction for rapid low resource asr development. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 258–265. IEEE.

Luis Fernando D'Haro and Rafael E Banchs. 2016. Automatic correction of asr outputs by using machine translation.

Anirudh Mani, Shruti Palaskar, Nimshi Venkat Meripo, Sandeep Konam, and Florian Metze. 2020. Asr error correction and domain adaptation using machine translation. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Zhong Meng, Zhuo Chen, Vadim Mazalov, Jinyu Li, and Yifan Gong. 2017. Unsupervised adaptation with domain separation networks for robust speech recognition. *arXiv preprint arXiv:1711.08010*.

Bodenreider O. 2004. The unified medical language system (umls): integrating biomedical terminology. Nucleic Acids Res. 2004 Jan 1;32(Database issue):D267-70. doi: 10.1093/nar/gkh061. PubMed PMID: 14681409; PubMed Central PMCID: PMC308795. Nucleic Acids Res.

Vijayaditya Peddinti, Guoguo Chen, Vimal Manohar, Tom Ko, Daniel Povey, and Sanjeev Khudanpur. 2015. Jhu aspire system: Robust lvcsr with tdnns, ivector adaptation and rnn-lms. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 539–546. IEEE.

Sai P Selvaraj and Sandeep Konam. 2019. Medication regimen extraction from medical conversations. *arXiv preprint arXiv:1912.04961*.

Yusuke Shinohara. 2016. Adversarial multi-task learning of deep neural networks for robust speech recognition. *Proc. Interspeech 2016*.

Prashanth Gurunath Shivakumar, Haoqi Li, Kevin Knight, and Panayiotis Georgiou. 2019. Learning from past mistakes: improving automatic speech recognition output via noisy-clean phrase context modeling. *APSIPA Transactions on Signal and Information Processing*, 8.

Luca Soldaini and Nazli Goharian. 2016. Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR workshop, sigir*, pages 1–4.

Sining Sun, Binbin Zhang, Lei Xie, and Yanning Zhang. 2017. An unsupervised deep domain adaptation approach for robust speech recognition. *Neurocomputing*, 257:79–87.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Pawel Swietojanski and Steve Renals. 2014. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 171–176. IEEE.

Pawel Swietojanski and Steve Renals. 2015. Differentiable pooling for unsupervised speaker adaptation. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4305–4309. IEEE.