Towards NLU Model Robustness to ASR Errors at Scale

Yaroslav Nechaev
work@remper.ru
Amazon Alexa
Cambridge, MA, USA

Weitong Ruan
weiton@amazon.com
Amazon Alexa
Cambridge, MA, USA

Imre Kiss
ikiss@amazon.com
Amazon Alexa
Cambridge, MA, USA

ABSTRACT

In a large-scale Spoken Language Understanding system, Natural Language Understanding (NLU) models are typically decoupled, i.e, trained and updated independently, from the upstream Automatic Speech Recognition (ASR) system that provides textual hypotheses for the user's voice signal as input to NLU. Such ASR hypotheses often contain errors causing severe performance degradation as the downstream NLU models are trained on clean human-annotated transcripts. Furthermore, as the ASR model updates, the error distribution drifts making it even harder for NLU models to recover and making manual annotation of erroneous ASR hypotheses impractical.

In this paper, we investigate data-efficient techniques applicable to a wide variety of NLU models employed in large-scale production environments to make them robust to ASR errors. We measure the effectiveness of such techniques as both the ASR error distribution and usage patterns change over time.

CCS CONCEPTS

• **Computing methodologies → Speech recognition**; **Supervised learning**; *Neural networks*.

KEYWORDS

datasets, natural language understanding, data augmentation, adversarial training, data-efficient learning

ACM Reference Format:
Yaroslav Nechaev, Weitong Ruan, and Imre Kiss. 2021. Towards NLU Model Robustness to ASR Errors at Scale. In *2nd International Workshop on Data-Efficient Machine Learning @KDD2021, 14–18 Aug, 2021, Virtual Conference.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.0000/0000000.0000000

1 INTRODUCTION

Spoken Language Understanding (SLU) systems quickly became a staple part of people's day to day activities. Cars, personal computers, phones and other consumer devices are equipped with personal assistants responding to voice commands. Behind each of these products there is an SLU system, most widely used of which, to name a few, are Siri, Google Assistant, Alexa and Cortana. Implementing SLU with comparable functionality requires significant effort from many engineers and scientists. To enable efficient participation of so many people in the development process and support fast model iteration, such SLU systems are typically decoupled into a multitude of interacting components, each solving its separate task. Thus, recent push in the academic community [8, 12, 17] towards powerful end-to-end SLU approaches may not be practical to adopt for complex SLU systems.

Two major parts of a typical SLU system are the Automatic Speech Recognition (ASR) and the Natural Language Understanding (NLU) components. ASR is tasked with performing necessary processing to convert user's speech into text. Natural Language Understanding (NLU) component [14] works on the output of the ASR system predicting a domain (Domain Classification task or DC), intent (Intent Classification – IC) and performing slot filling (Named Entity Recognition or NER). For example, for utterance "what is the temperature in Barcelona" a typical pipeline could output `domain:weather`, `intent:get_weather` and the following NER annotation:

$$\underset{\text{O}}{\underline{\text{what is}}}\ \underset{\text{request}}{\underline{\text{the temperature}}}\ \underset{\text{O}}{\underline{\text{in}}}\ \underset{\text{location}}{\underline{\text{Barcelona}}}.$$

Due to this separation, ASR errors propagate to downstream NLU components at times causing severe performance degradation [14]. If NLU component is trained and evaluated solely on clean human-annotated transcripts then it will have little to no chance to recover from such errors. Additionally, NLU models trained to capture semantic similarity between words can have a hard time dealing with phonetic ambiguity, which is the main cause of ASR errors. Thus, measures have to be implemented to make NLU systems robust to upstream errors. Recently this problem has attracted considerable interest from researchers proposing a wide variety of approaches to mitigate this drop in performance [6, 14, 21].

Additional challenges arise when trying to solve the issue of ASR errors in a large-scale SLU system. Firstly, the ASR component is periodically updated with the new releases causing a drift in both the error distribution and the distribution of any auxiliary signals, such as confidence scores. Thus, a system trained to be robust to errors produced by a particular release of the ASR system can fail to adjust to the newer versions. Secondly, the NLU itself can be decoupled with multiple models of varying complexity being deployed at the same time, requiring a potential robustness solution to be applicable to as many model families as possible with little or no modification. In third, the usage patterns change in response to new trends and new features which, compounded with periodic ASR releases, further exacerbates the input distribution drift. Finally, real-world SLU systems can have strict constraints on both the inference and the training times to maintain reasonable operational costs.

In this paper, having these challenges in mind, we investigate three simple techniques to improve downstream model robustness to ASR

DeMaL2021@KDD2021, 14–18 Aug, 2021, Virtual Conference
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.0000/0000000.0000000

errors: Data Augmentation, Adversarial Training and a Confidence-Aware Layer. We design these approaches to be as data-efficient, lightweight and model-agnostic as possible to be applicable to a wide selection of NLU models while introducing as little overhead to production latency and operational costs as possible. We showcase this flexibility by applying them to a number of domain classification (DC) models of varying complexity.

Then, we measure the performance of the proposed approaches in presence of ASR errors. To this end, we employ different snapshots of our internal test and training sets covering roughly a one year period. This allows us to measure the effectiveness of these techniques as the input distribution starts to drift due to new ASR model releases and changes in usage patterns. We employ the following evaluation protocol which we suggest to be adopted both by industry and academia in future to ensure model robustness to ASR errors in presence of various distribution drifts. We measure the proposed and the baseline NLU model performance using both the human-annotated transcripts and the ASR hypotheses as input. A successful model is expected to show both (i) significant improvement as measured on ASR hypotheses and (ii) absence of degradation as evaluated on transcripts. The former gives us a measure of model robustness to ASR errors produced by a specific ASR system, while the latter serves as a proxy measure to the ability of the algorithm to generalize and potentially adapt to distribution drifts.

The rest of the paper is organized as follows. In Section 2 we summarize related work, while Section 3 describes proposed robustness approaches. Section 4 defines models, datasets and metrics used to measure robustness performance. Section 5 provides in-depth evaluation of the proposed approaches. Section 6 concludes the paper.

2 RELATED WORK

Various previous works tackled the problem of model robustness to ASR Errors. A number of data augmentation techniques were proposed exploiting ASR hypotheses [3, 11]. Further improvements proposed to leverage word confusion information stored in the Word Confusion Network (WCN) [5, 7, 18, 19]. In [7], the authors proposed a LatticeRNN approach which builds an embedding from the lattice generated by the ASR. However, these approaches are data-intensive and time-consuming since, as the ASR model improves over time, previously annotated ASR one-best hypotheses become stale and cannot be used for training. To avoid this problem, Simonnet et al. [15] proposed an approach to simulate ASR errors from transcriptions and showed that the noising process helps to improve the robustness of the SLU system to ASR errors especially in case of insufficient training data.

A number of studies aimed at improving the representation learning instead. Following the success of learning continuous word representation, e.g. *word2vec*, in [13, 14] the authors came up with a *confusion2vec* [2], where the output embeddings contain not only semantic and syntactic relations of words in human language but also acoustic relationship between words. Similarly, Chung and Glass [2] designed a *speech2vec* architecture based on a RNN Encoder-Decoder framework and a skipgram or continuous bag-of-words training methodology. The resulting embedding has the capacity of capturing speech signal not in plain text. Huang and Chen [6] proposed to add a confusion loss to the task loss on top of the Embeddings from Language Model (ELMo) model during the fine-tuning stage. This confusion loss penalizes a negative cosine distance between the Language Model (LM) representations of each confusion word pairs, forcing the language model to generate similar contextualized embedding representation for the confusion word pairs. In Zhu et al. [21], the problem was approached from the perspective of training and evaluation mismatch. The proposed approach uses one layer of shard BiLSTM encoder and three task-oriented BiLSTM decoders. During model training, it forms a multi-task learning problem where the first task is a slot-tagging task over annotated transcription; the second is an unsupervised reconstruction task for both transcript and ASR hypotheses; the third is another task-invariant task where for each intermittent hidden output from the shared encoder layer, another FNN is used to classify them as either label or ASR hypotheses or transcription. The experimental results suggest that with the extra loss, the learned representation makes the model robust to ASR errors.

There are also approaches that aims to solve this problem from the ASR perspective. Soni et al. [16] proposed to model additive noise and channel mismatch distortion using a parametric generative model. They demonstrated that their proposed approach reduces the word error rate for ASR in unseen conditions.

Finally, our Data Augmentation and Adversarial Training implementation is closely related to the one proposed in Ruan et al. [11]. This paper is a followup investigation focusing on deployment considerations of these approaches in a large-scale SLU system.

3 MODEL ROBUSTNESS TO ASR ERRORS

A large NLU system usually consists of three statistical models: a domain classifier (DC), intent classifier (IC) and a named entity recognition (NER) components. As mentioned above, in this paper we specifically focus on DC robustness, however, approaches described here can be applied with little or no modification to IC or any other classification task. Some modifications will be required for NER component [11].

Here we detail three techniques that we used to improve DC robustness to ASR errors: Data Augmentation, Adversarial Training and Confidence-Aware Layer. These approaches were designed with the following production considerations in mind: (i) be data-efficient: no additional requirements for data annotation should be imposed and the approach should not significantly degrade training and inference time, (ii) be applicable to a wide range of models in a plug and play fashion.

Our Data Augmentation (DA) is a relatively straightforward approach: for utterances in the training set we add the corresponding 1-best hypotheses from the ASR output with the same domain label. DA is completely model-agnostic and can be applied during the data pre-processing step, which makes implementing it completely transparent to model builders in a large organization. The main design consideration for DA is the selection of the source of the ASR hypotheses. As the ASR component evolves, the error distribution shifts, potentially making NLU model trained on the old distribution less effective. Updating the entire training set every time the ASR is changed can be impractical as it is both costly to perform and takes time. To avoid this issue, we show the effect of different ASR

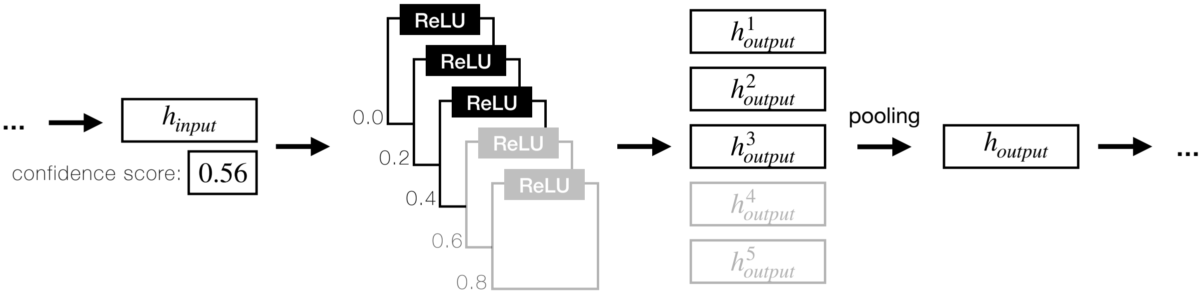

Figure 1: Overview of the Confidence-Aware Layer architecture.

update strategies on model performance. Our models were trained with an early stopping criteria on a validation set. Curiously, both the baseline and the augmented system took comparable number of updates to converge, making DA training time similar to baseline. Note that hypotheses beyond the 1-best can also be used to augment the training set, however, the more noisy the hypotheses, the more likely it is to change the semantics of the input utterance and cause generalization problems during training.

To allow for the greater control over the model behavior a generalization of DA called Adversarial Training (AT) was proposed [11]. Inspired by recent Adversarial Training approaches the idea is to consider ASR errors as a source of input perturbations. Formally, given the loss function $\mathcal{L}(y_{true}, y_{pred})$ for the target model, where y_{true} is a true label, y_{pred} is a predicted label given the input x_{pred} the resulting objective function will be defined as follows:

$$J = \beta\big(\mathcal{L}(y_{true}, y_{trans}) + \gamma\mathcal{L}(y_{true}, y_{asr})\big) + \alpha D_{KL}(y_{trans}||y_{asr})$$

where $\beta = \frac{1-\alpha}{1+\gamma}$ and $\alpha \in [0, 1)$. y_{trans} and y_{asr} are target model predictions given respectively the transcription x_{trans} and the corresponding ASR hypothesis x_{asr} as input; D_{KL} is a KL-divergence. KL-divergence term ensures consistency of predictions for the same utterance regardless of whether it is based on a clean transcription or on ASR output containing an error. Note that when $\alpha = 0$ this approach turns into DA. Parameter γ is either set to 1 or equal to the ASR confidence score for the corresponding input sample.

In the above formulation semantics of the input utterance can change due to an ASR error: user saying "bye iphone" can be misheard as "buy a phone", yielding completely different domain in DC. Thus, having the same label when computing $\mathcal{L}_{asr}$ and D_{KL} can potentially cause performance degradation. This is akin to label noise problem and, to achieve robustness to such noise, we have tested a variation of AT replacing the categorical cross entropy (CCE) loss for the ASR hypothesis with a Q-loss [20], a generalization of CCE and the mean-absolute error (MAE) losses, that allows controlling noise-robustness properties of a loss. Formally:

$$\mathcal{L}_q(y_{true}, y_{asr}) = y_{true}\frac{(1 - y^q_{asr})}{q} \qquad (1)$$

parameter $q \in [0, 1]$, which controls the balance between noise-resistant MAE (achieved when $q = 1$) behaviour and CCE (approximated as $q \to 0$), can be set depending on the ASR confidence score (c_{asr}) like so: $q = (1 - c_{asr})^{q'}$, we refer to this variation as Adaptive Q-loss.

Just like DA, Adversarial Training does not degrade inference performance and is applicable to a wide range of statistical models in a plug-and-play fashion due to being agnostic to the model structure. However, some modification of the model and the training workflow were required to adopt AT making it slightly harder to productize compared to DA. Similarly to DA, AT does not significantly affect training time, however, we did observe some performance degradation due to required realignment of transcriptions and ASRs during the computation of the loss. Note that in all cases the ASR confidence score is only used during the training. As the ASR component updates, confidence score distributions can change as well potentially making the ASR confidence score an unreliable feature.

Finally, Confidence-Aware Layer (CA) is simple modification aimed at exploiting utterance-level ASR confidence score (c_{asr}) during training and optionally during inference. Figure 1 provides an overview of the proposed approach. CA is applied to an existing layer in a neural network (in picture – a regular fully-connected layer with ReLU activation function), duplicating target layer n times forming "buckets", each corresponding to the specific c_{asr} threshold. Then, during the forward pass only the buckets that correspond to thresholds that are lower than the input c_{asr} are activated. The resulting outputs are pooled using either maximum, average or attention. Note that with this formulation each bucket receives all available transcripts (which have $c_{asr} = 1.0$) and progressively more noisy ASR hypotheses as threshold is reduced. Pooling such buckets create an ensemble effect similar to pooling multiple data-augmented models with different amounts of ASR hypotheses added into the training set. Threshold values and the number of buckets can be varied: we used 5 and 10 buckets, selecting either uniformly distributed threshold values or split to achieve roughly the same number of ASR hypotheses for each resulting score interval. Such technique can be applied to any layer or groups of layers in the network, however, when applied to a fully-connected layer, a very efficient implementation is possible allowing computing all bucket's transformations in a single matrix operation, making CA equivalent to a single wider layer. CA can be used without a confidence score during inference by activating all buckets together, avoiding adding the runtime dependency on c_{asr}.

4 EXPERIMENTAL SETTING

In this section, we describe the datasets and metrics we used in our experiments as well as define domain classification models that were used as targets for our robustness approaches. All experiments were done on our internal data and designed to be as close to production pipeline as possible to reflect real challenges that arise from upstream ASR errors in a large-scale NLU. Following evaluation criteria defined above, we evaluate both on the human-annotated transcriptions and on the ASR 1-best hypotheses aiming to balance model generalization with robustness to a specific ASR input distribution.

4.1 ASR hypotheses

In our experiments we use ASR hypotheses and the associated utterance-level confidence scores to augment the target models. To showcase model performance, we use two datasets including both transcriptions and ASR 1-bests: ASR-5M and ASR-9M. The former includes utterances covering four years and includes 5.5M utterances. The latter represents utterances covering a one year period after ASR-5M, has 9M samples and has significantly lower word error rate in its ASR hypotheses.

To perform distribution shift analysis in the later stages of our experiments we have produced two additional datasets: ASR-19a and ASR-19b, representing utterances coming from two fixed production ASR engine releases roughly half a year apart. ASR-19a+b is a combination of the two.

In production NLU systems lots of synthetic textual data is often used to assist in development of new features and for such utterances we would not have the corresponding ASR hypotheses. In our case, the datasets described above cover roughly a third of the training set depending on a domain and training setup. To mitigate this issue, we employ a recent text-level technique for simulating ASR errors [4] to produce ASR hypotheses for synthetic utterances mimicking real error distribution and word error rate. This procedure completes the augmentation of the training set yielding the ASR-ALL dataset containing 29M samples.

4.2 Metrics

To measure the overall NLU performance we use Semantic Error Rate (SEMER) defined as follows:

$$\text{SEMER} = \frac{N_{\text{errors}}}{N_{\text{reference slots}}} = \frac{N_S + N_I + N_D}{N_S + N_D + N_C}, \qquad (2)$$

where N_S, N_I, N_D and N_C are the number of substitution, insertion, deletion errors and the correct slots respectively. This metric is similar to Slot Error Rate (SER) employed in information extraction [10]. When assessing the DC component performance separately, we use per-domain error rate (fDCER) defined as $1 - F1$. To aggregate the performance across all domains, we calculate DC error rate defined as the corresponding weighted average: $1 - F1_{macro}$.

4.3 Models

We employ five different models of increasing complexity to evaluate robustness approaches. In our experiments, DC models are binary classifiers for each individual domain. Domain prediction is then acquired in a one-vs-all fashion.

BOW uses a simple logistic regression on top of sparse vector comprised of unigram, bigram and trigram features and serves as the simplest model in our tests. *FFNN* is a shallow neural model using fastText [1] embeddings as input followed by two layers of size 256 and ReLU activation functions. An *Ensemble* of these two classifiers is used combining the BOW and embedding-based features in a single network.

Table 1: Relative change in SEMER when DA is applied to each of the models. Performance on transcripts and ASR 1-bests is reported using ASR-5M and ASR-9M datasets respectively. Lower the better.

Model	ASR-5M (SEMER)		ASR-9M (SEMER)	
	Transcripts	ASR	Transcripts	ASR
BOW	+14.05%	-3.94%	+0.73%	-3.00%
FFNN	+2.52%	-4.98%	-0.75%	-4.14%
Ensemble	+6.34%	-5.19%	-0.29%	-3.75%
LSTM	+0.09%	-6.37%	–	–
LSTMCNN	-0.58%	-7.53%	-0.29%	-3.44%

Table 2: Performances (fDCER and SEMER) of the proposed approaches on transcripts and ASR 1-bests compared to Data Augmentation (DA) using ASR-ALL dataset. AT parameters: $\alpha = 0.3$ and $\gamma = 1$. CA uses uniform split on confidence score. Lower the better.

Approach	fDCER		SEMER	
	Transcripts	ASR	Transcripts	ASR
DA	–	–	–	–
AT	-0.56%	-0.78%	-0.50%	-0.35%
CA	-3.91%	-1.95%	-1.41%	-0.76%
DA+AT+CA	**-4.75%**	**-3.70%**	**-1.81%**	**-0.81%**

The forth system, *LSTM*, has two bi-directional LSTM layers with hidden state of size 256. Finally, we employ *LSTMCNN*, inspired by recent sequence tagging approach [9], as the most complex model in our tests. *LSTMCNN* exploit both character-level and word-level information via the combination of convolutional and recurrent layers. First, character embeddings (of size 16) are aggregated into word-level representations using a convolutional layer (with hidden state of size 32). Then, they are concatenated with an additional word vector and fed to two Variational LSTM layers (with hidden state of size 768).

5 EVALUATION

To assess the impact of the proposed robustness approaches we evaluate our models both on clean human-annotated transcripts and on the ASR 1-best hypotheses. Ideally, we would want to show an improvement on ASR hypotheses while maintaining performance on transcripts as a proxy measure for generalization allowing our models to show reasonable performance without the mandatory retraining as the ASR model updates.

Firstly, we evaluate Data Augmentation (DA) on models introduced above on datasets covering two time periods. Then, we compare Data Augmentation against Adversarial Training and the Confidence-Aware Layer approaches. Finally, we dive deep into practical considerations of applying such robustness techniques by (i) investigating the performance on different NLU domains, (ii)

Table 3: Data Augmentation performance (fDCER) against the baseline (non-augmented model) in the presence of the error distribution drift for different domains. Same ASR release as in ASR-19b is used for ASR evaluation. Relative improvements against the baseline are reported. Lower the better.

Approach	Across 3 domains		Global		Knowledge		Shopping	
	Trans	ASR	Trans	ASR	Trans	ASR	Trans	ASR
Baseline	–	–	–	–	–	–	–	–
DA (ASR-19a)	-0.04%	-1.96%	-0.24%	-1.51%	-1.56%	-2.28%	-2.33%	-2.39%
DA (ASR-19b)	-0.57%	-2.78%	-0.39%	-2.39%	-0.73%	-3.00%	-1.59%	-3.70%
DA (ASR-19b, $c_{asr} > 0.2$)	**-1.33%**	**-2.81%**	-0.60%	-2.60%	-2.97%	-3.37%	-2.29%	-3.25%
DA (ASR-19a+b)	+1.00%	-1.81%	+1.99%	-1.60%	+3.57%	-1.51%	-0.18%	-4.92%

studying the effect of ASR error distribution drift on different ASR hypothesis update strategies.

5.1 Data Augmentation

To evaluate Data Augmentation approach, we setup a set of controlled experiments assessing its impact on the NLU performance using different DC models. IC and NER models are left fixed throughout our experiments, always making their predictions based on clean transcripts for each utterance: this way in this experiment we effectively consider IC and NER to be perfectly robust to ASR errors to further highlight the DC performance. To showcase the effect of overtime improvements in ASR and the training set design we include evaluation results for two disjoint time periods: ASR-5M and ASR-9M (Table 1).

In ASR-5M setting, as the models are getting more sophisticated, SEMER improves in absolute terms but the gap stays the same. When Data Augmentation is employed, all models are exhibiting significant improvements ranging from 4% to 7.5% SEMER on ASR 1-best. However, performance on transcripts is a different story: less sophisticated models (*BOW*, *FFNN* and *Ensemble*) there show significant performance degradation hinting at such models' limited ability to generalize. *LSTMCNN*, being the most complex model in our testing, was able to benefit the most from the Data Augmentation yielding 7.53% improvement on ASR 1-bests and 0.58% improvement on transcripts. Evaluation for the ASR-9M setting paints a similar picture: Data Augmentation was able to outperform baselines across the board both on transcripts and ASR 1-bests. Smaller improvements can be attributed to lower word error rate for ASRs in this dataset.

For the remainder of the paper *LSTMCNN* is adopted as a baseline for robustness modifications.

5.2 Advanced approaches

To evaluate the techniques described in Section 3, we employ the ASR-ALL dataset to augment the training set and use the same test sets as for ASR-9M for testing. Here we report two sets of results: the overall SEMER impact across the NLU stack and the fDCER numbers to isolate impact on just the DC component. Differently from the previous experiment, IC and NER components will also receive ASR hypotheses as input. This means that even if DC prediction is correct, the overall performance can still suffer in case the downstream task (IC or NER) is not able to recover from the ASR error giving us a complete picture on the effect of our modifications on the overall NLU stack.

As can be seen in Table 2, each of the approaches provide additional performance benefits on top of the Data Augmentation with the combination of all techniques exhibiting the best performance. Less pronounced SEMER improvements is due to IC and NER dampening the overall NLU performance. Q-loss was expected to perform well, however, in our experiments we did not observe statistically significant improvements ($q = \{0.1, 0.01\}$ both in constant and adaptive settings). For CA, choice of thresholds significantly affected per-domain performance. In terms of pooling strategy, average worked the best, while attention showed overall performance degradation.

5.3 Distribution drift analysis

Finally, in Table 3 we study how Data Augmentation performs for different domains in presence of error distribution drift due to the ASR component updates. Specifically, we used the test set with ASR hypotheses coming from the same ASR release as the ASR-19b dataset, while the training set was augmented with either "old" (ASR-19a) or "new" (ASR-19b) data or a combination of the two. In almost all cases, ASR augmentations allowed for improved performance as evaluated on ASR. Only the old hypotheses (ASR-19a) have caused overall degradation on ASR even though performance improvements on the three selected domains were observed. Due to this, we recommend updating ASR hypotheses periodically as practical. On transcripts, the combination of two ASR releases (ASR-19a+b) have shown degradation, warning about potential issues when combining different error distributions. Filtering out noisy ASR hypotheses ($c_{asr} > 0.2$) yielded much greater stability on transcripts without loss of ASR performance. This behaviour persisted when we evaluated across all domains beyond the three reported — **-1.11%** improvement on transcriptions and **-2.81%** on ASR hypotheses.

6 CONCLUSIONS

In this paper we tackle a problem of NLU model robustness to upstream ASR errors in a large-scale decoupled SLU system. To this end, we tested three techniques: Data Augmentation, Adversarial Training and Confidence-Aware Layer. These techniques are designed to be scalable, i.e. without major training and inference penalties and applicable to a wide range of machine learning models. We tested these approaches in different scenarios, such as, ASR error distribution drift, usage changes as reflected in NLU training and test sets across different years and different recipient models. While the results differ significantly from one scenario to another changing the expected benefit depending on the quality of ASR engine and the base model, in most cases we report meaningful performance improvements both as evaluated on ASR hypotheses and on annotated transcripts. These results demonstrate that for sufficiently complex model robustness to ASR errors is achievable without overfitting to a particular ASR error distribution. However, as the ASR engine updates, periodic update of the ASR hypotheses to mitigate the effect of the distribution drift is becoming more important.

In future, we will continue experimenting with other techniques applicable to large-scale SLU systems as well as covering the rest of the NLU components.

REFERENCES

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[2] Yu-An Chung and James R. Glass. 2018. Speech2Vec: A Sequence-to-Sequence Framework for Learning Word Embeddings from Speech. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana (Ed.). ISCA, 811–815. https://doi.org/10.21437/Interspeech.2018-2341

[3] Anoop Deoras and Ruhi Sarikaya. 2013. Deep belief network based semantic taggers for spoken language understanding. In *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, Frédéric Bimbot, Christophe Cerisara, Cécile Fougeron, Guillaume Gravier, Lori Lamel, François Pellegrino, and Pascal Perrier (Eds.). ISCA, 2713–2717. http://www.isca-speech.org/archive/interspeech_2013/i13_2713.html

[4] Maryam Fazel-Zarandi, Longshaokan Wang, Aditya Tiwari, and Spyros Matsoukas. 2019. Investigation of Error Simulation Techniques for Learning Dialog Policies for Conversational Error Recovery. *CoRR* abs/1911.03378 (2019).

[5] Dilek Hakkani-Tür, Frédéric Béchet, Giuseppe Riccardi, and Gökhan Tür. 2006. Beyond ASR 1-best: Using word confusion networks in spoken language understanding. *Comput. Speech Lang.* 20, 4 (2006), 495–514. https://doi.org/10.1016/j.csl.2005.07.005

[6] Chao-Wei Huang and Yun-Nung Chen. 2019. Learning ASR-Robust Contextualized Embeddings for Spoken Language Understanding. *CoRR* abs/1909.10861 (2019). arXiv:1909.10861 http://arxiv.org/abs/1909.10861

[7] Faisal Ladhak, Ankur Gandhe, Markus Dreyer, Lambert Mathias, Ariya Rastrow, and Björn Hoffmeister. 2016. LatticeRnn: Recurrent Neural Networks Over Lattices. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, Nelson Morgan (Ed.). ISCA, 695–699. https://doi.org/10.21437/Interspeech.2016-1583

[8] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech Model Pre-Training for End-to-End Spoken Language Understanding. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, Gernot Kubin and Zdravko Kacic (Eds.). ISCA, 814–818.

[9] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1064–1074. https://doi.org/10.18653/v1/P16-1101

[10] John Makhoul, Francis Kubala, Richard Schwartz, Ralph Weischedel, et al. 1999. Performance measures for information extraction.

[11] Weitong Ruan, Yaroslav Nechaev, Luoxin Chen, Chengwei Su, and Imre Kiss. 2020. Towards an ASR Error Robust Spoken Language Understanding System. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, Helen Meng, Bo Xu, and Thomas Fang Zheng (Eds.). ISCA, 901–905.

[12] Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. 2018. Towards End-to-end Spoken Language Understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 5754–5758.

[13] Prashanth Gurunath Shivakumar and Panayiotis G. Georgiou. 2019. Confusion2Vec: towards enriching vector space word representations with representational ambiguities. *PeerJ Computer Science* 5 (2019), e195. https://doi.org/10.7717/peerj-cs.195

[14] Prashanth Gurunath Shivakumar, Mu Yang, and Panayiotis G. Georgiou. 2019. Spoken Language Intent Detection using Confusion2Vec. *CoRR* abs/1904.03576 (2019). arXiv:1904.03576 http://arxiv.org/abs/1904.03576

[15] Edwin Simonnet, Sahar Ghannay, Nathalie Camelin, and Yannick Estève. 2018. Simulating ASR errors for training SLU systems. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*, Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (Eds.). European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2018/summaries/827.html

[16] Meet Soni, Sonal Joshi, and Ashish Panda. 2019. Generative Noise Modeling and Channel Simulation for Robust Speech Recognition in Unseen Conditions. *Proc. Interspeech 2019* (2019), 441–445.

[17] Natalia A. Tomashenko, Antoine Caubrière, Yannick Estève, Antoine Laurent, and Emmanuel Morin. 2019. Recent Advances in End-to-End Spoken Language Understanding. In *Statistical Language and Speech Processing - 7th International Conference, SLSP 2019, Ljubljana, Slovenia, October 14-16, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 11816)*, Carlos Martín-Vide, Matthew Purver, and Senja Pollak (Eds.). Springer, 44–55.

[18] Gökhan Tür, Anoop Deoras, and Dilek Hakkani-Tür. 2013. Semantic parsing using word confusion networks with conditional random fields. In *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, Frédéric Bimbot, Christophe Cerisara, Cécile Fougeron, Guillaume Gravier, Lori Lamel, François Pellegrino, and Pascal Perrier (Eds.). ISCA, 2579–2583. http://www.isca-speech.org/archive/interspeech_2013/i13_2579.html

[19] Xiaohao Yang and Jia Liu. 2015. Using word confusion networks for slot filling in spoken language understanding. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 1353–1357. http://www.isca-speech.org/archive/interspeech_2015/i15_1353.html

[20] Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 8792–8802. http://papers.nips.cc/paper/8094-generalized-cross-entropy-loss-for-training-deep-neural-networks-with-noisy-labels

[21] Su Zhu, Ouyu Lan, and Kai Yu. 2018. Robust Spoken Language Understanding with Unsupervised ASR-Error Adaptation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 6179–6183. https://doi.org/10.1109/ICASSP.2018.8461831