ON THE ROBUSTNESS OF GOAL ORIENTED DIALOGUE SYSTEMS TO REAL-WORLD NOISE

Jason Krone, Sailik Sengupta, Saab Mansour
Amazon AWS AI
East Palo Alto, CA, USA
{kronej,sailiks,saabm}@amazon.com

ABSTRACT

Goal oriented dialogue systems in real-word environments often encounter noisy data. In this work, we investigate how robust these systems are to noisy data. Specifically, our analysis considers intent classification (IC) and slot labeling (SL) models that form the basis of most dialogue systems. We collect a test-suite for six common phenomena found in live human-to-bot conversations (abbreviations, casing, misspellings, morphological variants, paraphrases, and synonyms) and show that these phenomena can degrade the IC/SL performance of state-of-the-art BERT based models. Through the use of synthetic data augmentation, we are improve IC/SL model's robustness to real-world noise by +11.5% for IC and +17.3 points for SL on average across noise types. We make our suite of noisy test data public to enable further research into the robustness of dialog systems.

1 INTRODUCTION

Intent classification (IC) and slot labeling (SL) models have achieved impressive performance, reporting accuracies above 95% Chen et al. (2019) on public benchmarks such as ATIS Hemphill et al. (1990) and Snips Coucke et al. (2018). These datasets commonly used in the academia are clean, while real-world data is often noisy. It is important to evaluate how robust goal oriented dialogue systems are to commonly seen noisy inputs and, if necessary, improve their performance on noisy data. In this work, we identify and evaluate the impact of six noise types (casing variation, misspellings, synonyms, paraphrases, abbreviations, and morphological variants) on IC and SL performance. We collect a suite of realistic ATIS and SNIPS test data for these phenomena.

We find that noise reduces the IC/SL performance of state-of-the-art BERT based IC/SL models by an average of -13.5% for IC and -18.9 points for SL. Training augmentation largely offsets these losses, improving performance on noisy benchmarks by $+11.5\%$ for IC and $+17.3\%$ for SL F1 on average. Yet there remains substantial headroom on our test suite for further improvement on abbreviations, morphological variants, and synonyms.

In summary, our contributions are three-fold: (1) We publicly release a benchmarking suite of IC/SL test data for six noise types commonly seen in real-word environments[1]; (2) We quantify the impact of these phenomena on IC and SL model performance; (3) We demonstrate that training augmentation is an effective strategy to improve IC/SL model robustness to noisy text.

2 RELATED WORK

Machine translation (MT) literature demonstrates that both synthetic and natural noise degrade neural MT performance Belinkov & Bisk (2018); Niu et al. (2020). Karpukhin et al. (2019) show that training on synthetic noise improves the robustness of MT to natural noise. We pursue a CheckList-style approach Ribeiro et al. (2020), characterizing robustness to noise prevalent in production– this goes beyond considering a subset of noise types such as paraphrases Einolghozati et al. (2019) or misspellings Pruthi et al. (2019). These prior works, as well as concurrent work on robustness gym

[1] Please email the authors to obtain the noised test data.

Phenomena	Train Generation	Test Generation	Examples
Abbreviations	Rule Based	Human	book a flight from san jose *2* nyc
Casing	Rule Based	Rule Based	BOOK A FLIGHT FROM SAN JOSE TO NYC
Misspellings	Hasan et al. (2015)	Spell. Corrections	*buk* a flight from San Jose to NYC
Morph.	Internal DB + LM	Human	start *booking* a flight from san jose to nyc
Paraphrasing	Back Translation	Human	*can you book me* a flight from San Jose to NYC
Synonyms	WordNet + LM	Human	*reserve* a flight from San Jose to NYC

Table 1: Summary of methods to generate noisy data for training and testing purposes. As an example, we provide a noised version of the sentence ‘book a flight from San Jose to NYC’ for each category.

Goel et al. (2021), only evaluate on synthetically generated noise. A few works have collected realistic noisy benchmarks, but they do not present any strategy for improving the robustness of IC/SL models Peng et al. (2020). Further, only Einolghozati et al. (2019) considers robustness evaluation of IC and SL models; however, those experiments do not consider pre-trained language models which may provide some natural robustness to noise. In contrast, we employ expert data annotators to collect realistic noisy data, evaluate the impact of noise on state-of-the-art pre-trained language models, and present strategies to substantially improve model robustness to noise.

3 Noise Categories

Dataset	Phenom.	#Utt	#IC	#SL	#SV	BLEU
ATIS	Original	893	20	69	288	1.00
	Abbrev.	99	13	44	135	0.66
	Case.	893	20	69	288	0.00
	Morph.	115	15	45	154	0.59
	Para.	217	18	56	185	0.42
	Spell.	893	20	69	350	0.80
	Syn.	225	18	57	191	0.64
Snips	Original	700	7	39	1571	1.00
	Abbrev.	98	6	35	334	0.63
	Case.	700	7	39	1474	0.00
	Morph.	101	6	36	335	0.65
	Para.	197	6	36	585	0.54
	Spell.	700	7	39	1623	0.83
	Syn.	201	6	36	592	0.73

Table 2: Statistics of the noised test-sets– utterance (Utt), intent (IC), slot label (SL), and slot value (SV) counts as well as the average BLEU score between the original and noised utterances for ATIS and Snips.

We consider six types of noise that are prevalent in the traffic of a task oriented dialogue service – namely, misspellings, casing, synonyms, paraphrases, morphological variants, and abbreviations (see Table 1). With the exception of misspellings and casing, we employ trained data associates to collect test sets that are representative of naturally occurring noise. For the misspelling and casing phenomena, we automatically generate our test sets because high-quality generation is possible for English. In Table 2, we showcase the different statistics associated with the generated test sets. The last column highlights the average BLEU score between the utterances in the set of original utterances and the corresponding set of noised utterances for the different noise types. Casing has the highest impact on BLUE scores while paraphrasing and morphological variants, which can change multiple tokens and their positions, reduces the similarity of the noised utterance to the original test set utterance more than abbreviation, misspelling and synonyms, which are token-level noise types.

For the purpose of training augmentation, generating noisy data using human experts becomes expensive owing to the large quality of data needed to train neural models. Thus, we either propose new or leverage existing methods to automatically generate all phenomena. In this section, we describe the automatic generation procedure for each phenomena. Details on selection of the noise rate for augmenting training data is given in the appendix.

Casing. Casing variation is common in text modality human-to-bot conversations. Consider for example, the responses “john” vs. “John” vs. “JOHN” to a bot prompt “Can I have your first name?”. Given that the training data is mostly small-cased (for ATIS) and true-cased (for SNIPS), we evaluate the impact of capitalizing all characters in the test set; we simply capitalize all test utterances. For training augmentation, we inject all-caps noise into 50% of training tokens.

Model	ATIS				Snips			
	IC Acc.		SL F1		IC Acc.		SL F1	
	Orig.	Noisy	Orig.	Noisy	Orig.	Noisy	Orig.	Noisy
BERT	98.8	97.3	95.6	87.0	99.0	98.7	96.3	91.2
+MLM Aug.$^{5\%}$	**98.7**	97.4	**95.6**	86.8	98.8	98.4	**96.6**	91.9
+Train Aug.$^{20\%}$	**98.7**	**97.8**	95.5	**94.3**	**99.1**	**98.8**	96.4	**95.3**
+MLM & Train Aug.	98.6	97.6	95.5	94.5	99.0	**98.8**	96.0	95.1

Table 3: Robustness of BERT without and with robust training at pre-training time (MLM) and IC/SL fine-tuning time (Train.) to misspelling phenomena.

Misspellings. Misspelling test sets contain 15% misspelled words, sourced from a collection of public, human misspelling-correction pairs[2]. Using human misspelling pairs produces a more natural test set, but it does not generalize well to new languages or domains. Thus, for augmentation, we utilize a probabilistic approach to generate synthetic misspellings put forward in Hasan et al. (2015). This method introduces errors based on induction probabilities mined from natural data and bases character level edits on the QWERTY keyboard layout. We note that such misspellings in general differ from adversarial perturbations based on visual (eg. 'Book' → '8ook') Gao et al. (2018) or semantic (eg. 'book' → 'reserve') similarity Morris et al. (2020); Jin et al. (2020) but, can encompass a subset of adversarial edits– eg. 'book' → 'Bo0k' may be considered a misspelling due to the vicinity of the key '0' to 'O' on the US keyboard layout and in turn the existence of such misspellings in the English corpus.

Synonyms. We augment the training data with synonyms using a two step approach. First, we obtain a list of candidate synonyms via word net. Second, we introduce the synonym candidate that results in the lowest perplexity, as measured by DistilGPT2 Sanh et al. (2020), into the utterance. This perplexity filter ensures greater fluency.

Paraphrases. For training data augmentation, we considered using paraphrased generated using back-translation, from English to Chinese and back Mallinson et al. (2017); Einolghozati et al. (2019). For preserving slot-labels, we annotate the slot values in the back-translated text if they match the slot values in the original text, ignoring casing differences. In contrast to other noise types that are injected at the token-level, this is injected at the utterance level.

Morphological Variants. We generate morphological variants in a similar manner to synonyms. Namely, we first produce a list of candidate morphological variants, using an internal curated database of 120,000 morphological variant pairs sourced by human experts. We then select the candidate with the lowest perplexity.

Abbreviations. To synthetically construct abbreviations for augmenting the training data, we consider first a knowledge base of common abbreviations Beal (2021) and follow certain rule-based approaches to drop vowels from tokens (eg. 'people' → 'ppl') or include common abbreviation mappings between numerals and words which share phonetics (e.g., ”2” and ”to”).

4 Data Collection

We employ trained data associates to introduce synonyms, paraphrases, morphological variants, and abbreviations in ATIS and SNIPS test data. Due to cost constraints, we noise a subset of approximately 200 utterances for each dataset. We instruct the data associates to introduce the given phenomena into the carrier phrase portion of each utterance (i.e. the spans that do not correspond to slot values). In instances where the associates are unable to come up with a viable modification to an utterance, the utterance is excluded from the evaluation set. For this reason, our test sets for some phenomena, namely abbreviations and morphological variants, contain fewer than 200 utterances. We perform quality assurance on the collected data, using internal data specialist that ensure at least 95% of the examples in a sample containing 25% of each noisy test set are realistic and representative of the given noise type. In Table 2, we provide statistics for both the clean and noised data.

[2] https://www.dcs.bbk.ac.uk/~ROGER/corpora.html

5 Approach

We evaluate the robustness of a BERT based joint intent classification and slot labeling model, which is currently SOTA on the Snips and ATIS benchmarks Chen et al. (2019). Similar to Chen et al. (2019), we add an additional feed forward layers on top of the [CLS] and sub-token hidden representations to predict the IC and SL tags, respectively [3]. We use the cased BERT checkpoint pre-trained on the Books and Wikipedia corpora. We experiment to see the effectiveness of robust training at the fine-tuning stage and the pre-training stage for on misspelling noise and see that augmenting training data during the fine-tuning stage ensures higher robustness without increasing the cost of training (see Table 3; a detained description of these experiments can be found in subsection 6.1). Hence, we explore the use of training data augmentation in the fine-tuning phase for improved robustness.

6 Results

6.1 Robust Pre-training on Synthetic Misspellings

Noisy pre-training leverages recent breakthroughs in masked language modeling (MLM). We pre-train BERT on the Wikipedia and Books corpus augmented with synthetic misspellings at a rate of 5% for an additional 9,500 steps using the standard MLM objective. Noisy pre-training has the potential to be more efficient and generalize better than training augmentation. Ideally, a language model could be pre-trained once on a noisy corpus and subsequently adapted to any downstream task. In contrast, training augmentation must be applied separately to each downstream task, incurring greater computational cost as the number of tasks increase. Moreover, noisy pre-training on large corpora could improve generalization to noise not present in task specific training data, which may not be as large or diverse as the pre-training corpus. Empirically, we find that noisy pre-training does not perform as well as training augmentation. While noisy pre-training generates marginal improvements in ATIS IC accuracy (+0.1) and Snips SL F1 (+0.7), training augmentation produces substantially larger gains on the noised test sets. We utilize training augmentation, as opposed to pre-training augmentation, in experiments on all other phenomena as a result of this finding.

6.2 IC/SL Training Augmentation

We evaluate the impact of training augmentation on BERT IC/SL performance. We report IC accuracy and span level SL F1 scores, averaged over three random seeds, on parallel original (control) and noisy test splits (treatment) for each model setting in Table 4. An optimal model should close the gap between noisy and original performance without degrading original performance. In each four-cell of the table, we highlight the best possible IC accuracy or SL F1 value in **bold text**, indicate the drop in accuracy (if any) due to noisy data in red and highlight the gains obtained by training data-augmentation (if any) in green.

Abbreviations. Abbreviations degrade IC and SL performance on ATIS and Snips. This degradation is larger on ATIS (-9.5% IC) than Snips (-1.0% IC). ATIS abbreviations are often specific to the travel domain (e.g., “tix” vs. “tickets”). In contrast, Snips abbreviations are more general (e.g., “@” vs. “at”). We hypothesize that BERT is more likely to have trained on the abbreviations present in the Snips dataset than ATIS, and therefore is more robust on the Snips dataset. While augmentation boosts IC/SL performance on abbreviations to parity on Snips, a substantial gap in performance remains on ATIS. We find that the coverage of test abbreviations in the augmented training data is far lower for ATIS than for Snips (40% vs. 74%), which likely accounts for this difference.

Casing. Intent classification and slot labeling performance drop substantially on the noised test set because the Bert tokenizer fails to identify fully capitalized words in the vocabulary and instead breaks them down to match character-level sub-word tokens (eg. ‘would’ → {‘W’, ‘O’, ‘U’, ‘LD’}). Without the use of augmentation, where half-of-the training data is injected with capitalized form of words, the classifier is not able to associate these sub-token representations to intent classes or slot-labels. In general, cased BERT is not robust to the presence of fully capitalized strings as if

[3] Implemented using the gluon tutorial for intent classification and slot labeling: `https://gluon-nlp.mxnet.io/model_zoo/intent_cls_slot_labeling/index.html`

Phenomena	Model	ATIS				Snips			
		IC Acc.		SL F1		IC Acc.		SL F1	
		Orig.	Noisy	Orig.	Noisy	Orig.	Noisy	Orig.	Noisy
Casing	BERT	**98.8**	73.1	**95.6**	16.2	**99.0**	23.3	**96.3**	5.0
	+Aug.$^{50\%}$	98.7	98.1	94.6	94.9	98.9	98.8	96.0	90.0
Misspellings	BERT	**98.8**	97.3	**95.6**	87.0	99.0	98.7	96.3	91.2
	+Aug.$^{20\%}$	98.7	97.8	95.5	94.3	**99.1**	98.8	**96.4**	95.3
Abbreviations	BERT	96.0	86.5	**91.9**	75.9	97.9	96.9	95.8	85.2
	+Aug.$^{15\%}$	**96.6**	88.9	91.5	89.1	**98.3**	**98.3**	**96.6**	94.8
Morph.	BERT	**98.0**	97.1	92.3	93.5	98.0	96.9	95.8	85.2
	+Aug.$^{10\%}$	96.8	95.9	94.0	**94.1**	**98.3**	**98.3**	**97.2**	88.8
Synonyms	BERT	**96.9**	57.6	89.0	86.6	**98.8**	98.3	95.2	94.8
	+Aug.$^{10\%}$	97.2	90.4	**92.3**	90.2	99.2	98.7	**96.9**	96.4
Paraphrases	BERT	**97.1**	90.9	88.3	86.7	98.8	98.5	95.5	93.7
	+Aug.$^{15\%}$	**97.1**	90.5	**88.8**	86.8	**99.0**	98.8	**95.9**	94.4

Table 4: Robustness results across different phenomena for ATIS and SNIPS. For each phenomena, the baseline (BERT) and data augmentation (Aug.) with a specified percentage noise rate (x%) are compared using intent classification accuracy (IC Acc.) and slot labeling F1 (SL F1) scores.

fails to leverage the representation of larger sub-words in the vocabulary and behaves similar to a character-level model.

Misspellings. Test time misspellings do not impact IC accuracy more than 0.2 points because a misspelled word in the utterances only changes the sub-token breakdown of that word which in turn does not change the intent of the sentence ('what' *vs.* {'wa','t'}). While majority of the noise in the test set is in the carrier phrase tokens (as opposed to the tokens representing slot values), we notice a drop in SL F1 of -8.6 on ATIS and -5.1 on SNIPS. For atis, we note that slot values for slot labels such at period of day (eg. 'night' *vs.* 'nite') are more prone to misclassfication when injected with misspelled noise in comparison to slot values such as day names ('sunday' *vs.* 'suntday'). Augmentation of training data with 20% misspellings increases ATIS and SNIPS SL F1 score to within 1.2 points of performance on the original test set as the sub-tokens of misspelled words are now better recognized as slot values. This result demonstrates training on synthetic misspellings Hasan et al. (2015) can improve generalization to natural misspelling at test time.

Morphological Variants. Test time morphological variants lower Snips SL performance by 10.6% (from 95.8% to 85.2%). This loss in SL is distributed across error types, including hallucination of slots that are not present (30% of errors), prediction of the incorrect slot label (35% of errors), and prediction that no slot is present (35% of errors). In contrast, ATIS IC/SL and Snips IC are robust to morphological variation, deviating from original test scores by at most -1.1%.

Synonyms. Synonyms decrease ATIS IC performance by 39.3%, while the impact on Snips IC/SL is negligible. We note that the model picks up word to intent co-relations ('fare' → 'atis_airfare', 'flight(s)' → 'atis_flight', 'plane(s)' → 'atis_aircraft') and often when these words are replaced with a synonym, the model mis-classfies the intent. Further, changing the word 'flight(s)' to the word 'plane(s)' where they can be used as synonyms, the model's intent prediction flips from 'atis_flight' to 'atis_aircraft'. We conjecture that this difference in effect size is the result of over-fitting on less diverse ATIS carrier phrases. This lack of diversity is evidenced by the fact that ATIS contains roughly half as many unique carrier phrase tokens as Snips (430 vs. 842), despite having longer utterances on average (11.3 words vs. 9.1 words). Introducing synonyms into the ATIS training data boosts IC accuracy by +32.8% and SL F1 score by +3.6 points. On Snips, training augmentation performs comparably to the baseline system.

Paraphrases. Paraphrases lead to a moderate drop in ATIS IC performance (-6.2%) and a marginal drop in ATIS SL and SNIPS IC/SL (between -1.8 and -0.3). Similar to the cases of synonyms, we posit that ATIS IC is most impacted due to the lack of diverse carrier phrases in the training set and a greater degree of change between the original utterance and paraphrased version, demonstrated by 0.12 lower normalized BLEU score as compared to SNIPS. Training augmentation with back-translated paraphrases yields comparable scores to the baseline system.

7 Conclusion

In this paper, we investigate the impact of noise on dialogue systems. We demonstrate that SOTA BERT based IC/SL models are not robust to casing, synonyms, and abbreviations. While they are substantially more robust to morphological variation, paraphrases, and misspellings. We show the use of data augmentation improves performance by +11.5% IC and +17.3% SL F1 on average. Yet there remains a gap in noisy and original IC accuracy for synonyms and abbreviations, which emphasizes the need for novel techniques to improve model robustness. We hope that our benchmark will support future research in this direction and enable the design of more robust task oriented dialogue systems.

Acknowledgements The authors would like to thank the Amazon Lex team for engaging in insightful discussions and providing valuable feedback.

References

Vangie Beal. The complete list of 1500+ common text abbreviations & acronyms, 2021. URL https://www.webopedia.com/reference/text-abbreviations/.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *ICLR*, 2018.

Qian Chen, Zhu Zhuo, and Wen Wang. Bert for joint intent classification and slot filling, 2019.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190, 2018. URL http://arxiv.org/abs/1805.10190.

Arash Einolghozati, Sonal Gupta, Mrinal Mohit, and Rushin Shah. Improving robustness of task oriented dialog systems. *arXiv preprint arXiv:1911.05153*, 2019.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56. IEEE, 2018.

Karan Goel, Nazneen Rajani, Jesse Vig, Samson Tan, Jason Wu, Stephan Zheng, Caiming Xiong, Mohit Bansal, and Christopher Ré. Robustness gym: Unifying the nlp evaluation landscape. *arXiv preprint arXiv:2101.04840*, 2021.

Saša Hasan, Carmen Heger, and Saab Mansour. Spelling correction of user search queries through statistical machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 451–460, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1051. URL https://www.aclweb.org/anthology/D15-1051.

Charles T Hemphill, John J Godfrey, and George R Doddington. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8018–8025, 2020.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pp. 42–47, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5506. URL https://www.aclweb.org/anthology/D19-5506.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 881–893, Valencia, Spain, April 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/E17-1083`.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126, 2020.

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. Evaluating robustness to input perturbations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8538–8544, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.755. URL `https://www.aclweb.org/anthology/2020.acl-main.755`.

Baolin Peng, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. Raddle: An evaluation benchmark and analysis platform for robust task-oriented dialog systems. *arXiv preprint arXiv:2012.14666*, 2020.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5582–5591, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1561. URL `https://www.aclweb.org/anthology/P19-1561`.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*, 2020.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

A Appendix

A.1 Hyper Parameter Settings

We fine-tune BERT for up to 40 epochs with a batch size of 32. To prevent over fitting, we use early stopping on the validation loss. We optimize BERT parameters using gluonnlp's bertadam optimizer with a learning rate of 5e-5 and no weight decay. These are the default hyper-parameters provided in the gluon tutorial for intent classification and slot labeling (`https://nlp.gluon.ai/model_zoo/intent_cls_slot_labeling/index.html`).

A.2 Hyper Parameter Tuning

We use the default BERT joint IC/SL hyper-parameters mentioned above in all experiments, for both the baseline or training augmentation approaches. Using these fixed hyper-parameters, we tune the noise rate used for IC/SL training data augmentation. We tune the training data augmentation noise rate in the ranges listed below for each noise type. We select the noise rate that provides the best trade-off between performance on the control and treatment sets. The noise rates used in our final results are shown in the results table as a superscript.

Noise Rate Search Range by Noise Type:

- Casing: {50%, 100%}
- Misspellings: {10%, 15%, 20%, 25%, 30% }
- Abbreviations: {10%, 15%, 20%, 25%}
- Morphological Variants: {10%, 25%, 50%}
- Synonyms: {10%, 25%, 50%}
- Paraphrases: {5%, 10%, 15%}

A.3 Compute Environment

We run all experiments on p3.2xlarge GPU instances using the AWS Deep Learning AMI (Ubuntu 16.04).

A.4 Packages Used in Experimentation

We utilize the following packages to train and evaluate our models as well as generate synthetic noise for training augmentation:

- nltk==3.3
- gluonnlp==0.8.1
- mxnet==1.3.0
- numpy==1.14.3
- scikit-learn==0.23.1
- scipy==1.1.0
- torch==1.7.1
- transformers==4.2.2
- seqeval==0.0.12

A.5 Model Training and Inference Times

Training for 40 epochs plus inference on the test set takes approximately 11 minutes on the ATIS dataset and 21 minutes for the Snips data-set for both the baseline and training augmentation approaches.

Noise Type	Probability	Example	Char. Edit
Original	1.0 - p	list *flights* from las vegas to phoenix	N/A
Insertion	$p * 0.33$	list *fljights* from las vegas to phoenix	$+$j
Deletion	$p * 0.18$	list *flghts* from las vegas to phoenix	$-$i
Substitution	$p * 0.43$	list *flithts* from las vegas to phoenix	g $\rightarrow$ t
Transposition	$p * 0.06$	list *filghts* from las vegas to phoenix	l $\leftrightarrow$ i

Table 5: Instances of insertion, deletion, substitution, and transposition noise types for the example utterance (Original), "list flights from las vegas to phoenix". We sample each noise type with the given probability to construct noisy pre-training and IC/SL training datasets, where p is the noise rate. We inject noise into the token *"flights"* in each example and provide the character level edit (Char. Edit) that transforms the original token to the noised token.

A.6 Evaluation Metrics

We compute slot labeling F1 score using the seqeval library (`https://github.com/chakki-works/seqeval`). We utilize the evaluation function provided in the gluonnlp intent classification and slot labeling tutorial to compute intent classification accuracy (`https://nlp.gluon.ai/model_zoo/intent_cls_slot_labeling/index.html`).

A.7 Synthetic Misspelling Generation

We build on prior work by Hasan et al. (2015) to generate synthetic misspellings that are representative of natural misspellings. Hasan et al. present a taxonomy of misspelling types and induction probabilities mined from natural noise. This taxonomy consists of four noise types, substitution, insertion, deletion, and transposition.

Character choice in substitution and insertion operations is based on the QWERTY keyboard layout. Given a character c (e.g., ”d”), we substitute or insert a character the appears next to c on the QWERTY keyboard (e.g. ”f”, ”s”, ”e”, ”c”). We provide examples of each noise type and list the probability with which we introduce these types of noise in Table 5.