

# On the Robustness of Intent Classification and Slot Labeling in Goal-oriented Dialog Systems to Real-world Noise

Sailik Sengupta\*, Jason Krone\*, Saab Mansour

amazon AWS AI

{kronej, sailiks, saabm}@amazon.com

## Abstract

Intent Classification (IC) and Slot Labeling (SL) models, which form the basis of dialogue systems, often encounter noisy data in real-world environments. In this work, we investigate how robust IC/SL models are to noisy data. We collect and publicly release a test-suite for seven common noise types found in production human-to-bot conversations (abbreviations, casing, misspellings, morphological variants, paraphrases, punctuation and synonyms). On this test-suite, we show that common noise types substantially degrade the IC accuracy and SL F1 performance of state-of-the-art BERT-based IC/SL models. By leveraging *cross-noise robustness transfer* – training on one noise type to improve robustness on another noise type – we design aggregate data-augmentation approaches that increase the model performance across all seven noise types by +10.8% for IC accuracy and +15 points for SL F1 on average. To the best of our knowledge, this is the first work to present a single IC/SL model that is robust to a wide range of noise phenomena.

## 1 Introduction

Intent classification (IC) and slot labeling (SL) tasks (shown in Table 1) form the backbone of goal-oriented dialog systems. In recent times, deep-learning models have achieved impressive performance, reporting accuracies above 95% (Chen et al., 2019) on public IC/SL 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. Extensive prior work on robustness has focused on identifying noise-types that affect NLP systems and suggesting methods to improve robustness to individual noise types. However, two key questions remain – (1) what noise types are seen in real-world goal-oriented dialog systems,

Input Utterance	Play a song by <b>Bob Dylan</b>
Slot Labels (SL)	<b>O O O O</b> artist artist
Intent Class (IC)	<b>PlayMusic</b>

Table 1: Example input and outputs for IC & SL tasks.

and (2) how can we construct a single model robust to all real-world noise variants?

In this work, we identify and evaluate the impact of seven noise types frequently observed in a production – casing variation, misspellings, synonyms, paraphrases, punctuation, abbreviations, and morphological variants – on IC and SL performance. In this regard, 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 accuracy and –18.9 points for SL F1. We show that strategic augmentation and regularization approaches can offset these losses, improving performance on noisy benchmarks by +10.8% for IC and +15 points for SL F1 on average by leveraging cross-noise robustness transfer while minimally effecting accuracy on the original test-data. We emphasize that these improvements are for a single IC/SL model that is robust to all noise types. This is a substantial improvement in generalization over existing methods, which train separate models for each noise type.

To summarize our contributions:

- (1) We publicly release a bench-marking suit of IC/SL test data for seven noise types seen in real-world goal oriented dialog systems.<sup>1</sup>
- (2) We show that training data augmentation with one noise type (eg. synonyms) can improve robustness to unseen noise types (eg. paraphrases) substantially (eg. ATIS IC: +12.6); we call this *cross-noise robustness transfer*.

\* Equal Contribution.

<sup>1</sup>Dataset: [github.com/amazon-research/real-world-noisy-benchmarks-for-natural-language-understanding](https://github.com/amazon-research/real-world-noisy-benchmarks-for-natural-language-understanding) 

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)	Public DB	book a flite 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
Punctuation	(Tilk and Alumäe, 2016)	Human	book a flight from San Jose to NYC.
Synonyms	WordNet + LM	Human	reserve a flight from San Jose to NYC

Table 2: Summary of methods to generate noisy data for training and testing purposes. As an example, we provide a noised version of the sentence (in ATIS) ‘book a flight from San Jose to NYC’ for each category.

(3) We present a single IC/SL model that improves robustness to all seven noise types that we study by an average of +10.5% for IC accuracy and +15 points for SL F1.

## 2 Related Work

Prior works demonstrate the brittleness of neural systems in NLP to different noise types. Adversarial NLP focuses on construction of examples that impair a model’s performance (Jia and Liang, 2017; Ribeiro et al., 2018; Ebrahimi et al., 2017; Cheng et al., 2019). A parallel thread of research investigates the behavior of models towards real-word noise types. Machine Translation (MT) systems perform poorly in the presence of synthetic and natural character-level changes (eg. character swaps, typographical errors, misspellings) (Belinkov and Bisk, 2018; Karpukhin et al., 2019); similar conclusions hold for casing noise (Niu et al., 2020). Other works investigate robustness of NLP models to paraphrases (Einolghozati et al., 2019), misspellings (Pruthi et al., 2019), and morphological variants (Tan et al., 2020a). The most common approach to improving model robustness is data augmentation, either at pre-training or training time. For example, in (Tan et al., 2020a), augmentation with adversarial examples and a single epoch of pre-training can improve robustness of the model for Question Answering (QA) and MT tasks; a comprehensive summary of DA techniques can be found in (Feng et al., 2021). Recent works present frameworks for comprehensive model-agnostic testing of NLP systems to grammatical and noise variants (Ribeiro et al., 2020; Goel et al., 2021). All of these prior works focus on either (1) improving robustness to individual noise types or (2) simply evaluating model performance on synthetically-generated noise variants. In contrast, we construct a single model that is robust to all seven noise types seen in

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
	Misspl.	893	20	69	350	0.80
	Morph.	115	15	45	154	0.59
	Para.	217	18	56	185	0.42
	Punc.	243	14	49	212	0.68
	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
	Misspl.	700	7	39	1623	0.83
	Morph.	101	6	36	335	0.65
	Para.	197	6	36	585	0.54
	Syn.	201	6	36	592	0.73

Table 3: Statistics on 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.

real-world systems.

Beyond training-data augmentation, pre-training robust word-embeddings is a popular defense against adversarial examples (Zhu et al., 2020), misspellings (Piktus et al., 2019), and character ordering (Malykh et al., 2018). In (Kudo, 2018), authors investigate the impact of different sub-word tokenization techniques on model robustness. Effective regularization has also been shown to improve the robustness of IC/SL models to paraphrases (Einolghozati et al., 2019). Yet many of the techniques to improve robustness are not comparable due to evaluation on different test sets. Hence, we provide high-quality noisy test data to serve as a common test bed.

## 3 Noise Categories

We consider seven types of noise that are prevalent in the traffic of a task oriented dialogue service – namely, misspellings, casing, synonyms, paraphrases, punctuation, morphological variants, and abbreviations (see Table 2). 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 (1) high-quality generation is possible for English and (2) purposeful introduction of misspellings is not a natural task. Generating noisy data using human experts is expensive. Therefore, it is not suitable for the purpose of training augmentation. Thus, all of our training augmentation experiments rely upon automatically generated phenomena. In this section, we provide details about the manual data collection process and describe the automatic generation procedure leveraged for training data augmentation. In Table 3, we showcase statistics for ATIS and SNIPS noisy test sets.

### 3.1 Manual Test Data Collection

We employ trained data associates to introduce synonyms, paraphrases, morphological variants, and abbreviations in ATIS and SNIPS. We also use data associates to punctuate ATIS. ATIS requires special treatment for punctuation as it is unpunctuated, while SNIPS is punctuated and does not require manual collection for the punctuation noise type. 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.

### 3.2 Synthetic Data Generation

**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.

**Misspellings** Misspelling test sets contain 15% misspelled words, sourced from a collection of public, human misspelling-correction pairs<sup>2</sup>. 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.

**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.

**Punctuation** For ATIS, we augment the training data by using a bi-directional recurrent neural network with attention proposed in (Tilk and Alumäe, 2016). We note that the punctuation added to ATIS test sets are clearly noise or *irrelevant punctuation* (Ek et al., 2020) as the absence of these in the original test-sets does not impact intent or slot value classification. While some punctuation can be considered for complex scenarios in NLU (eg. use of comma to separate consecutive slot values), the idea of *relevant punctuation* is currently not well explored in the context of NLU systems. We do not experiment with the punctuation phenomena on the Snips dataset, since snips already contains punctuation in both the training and test splits.

**Morphological Variants** We generate morphological variants for training data augmentation in a similar manner to synonyms. Namely, we first

<sup>2</sup>[www.dcs.bbk.ac.uk/ROGER/corpora.html](http://www.dcs.bbk.ac.uk/ROGER/corpora.html) 

produce a list of candidate morphological variants, using an internal curated database of 120,000 morphological variant pairs sourced by human experts and then select the candidate with the lowest perplexity. While this is similar to the MORPHEUS approach described in (Tan et al., 2020b), it has two major differences. First, the list of morphed candidates in our case are chosen based on an internal curated database as opposed to the LemmInflect package<sup>3</sup>. Second, our approach is a model-agnostic approach that can work regardless of how the model is trained, whereas (Tan et al., 2020b) uses a model-in-the-loop approach to generate morphed inputs that are adversarial in nature (i.e. flip the model’s prediction).

**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 Approach

In this section, we highlight the various approaches we consider to construct a single model robust to various noise types seen in live traffic.

### 4.1 Model Setup

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.<sup>4</sup> We use the cased BERT checkpoint pre-trained on the Books and Wikipedia corpora.

### 4.2 Data Augmentation

**Training Data Augmentation (DA)** For each noise type, we consider augmenting the training data with noised versions of utterances in the original training set along-with the same intent and similar slot labels. For this purpose, we leverage the synthetic approaches described in subsection 3.2.

<sup>3</sup>[github.com/bjascob/LemmInflect](https://github.com/bjascob/LemmInflect)

<sup>4</sup>Implemented using the gluon tutorial: [nlp.gluon.ai/model\\_zoo/intent\\_cls\\_slot\\_labeling/index.html](https://nlp.gluon.ai/model_zoo/intent_cls_slot_labeling/index.html)

**Aggregate Data Augmentation (A-DA)** We explore two augmentation strategies that consider all noise types: *Uniform (Uni) Aggregation* and *Best Proportion (BP) Aggregation*. *Uni* augments the training set with 10% for each noise type (i.e. 10% abbreviation, 10% casing, 10% synonyms, etc.). *BP* tunes the training augmentation proportions based on the best performance observed on a validation set (15% abbreviation, 50% casing, 10% synonyms, etc.).

## 4.3 Regularization

### BERT-based Sub-word Regularization (B-SR)

There exists multiple options when splitting word tokens into sub-words. In (Kudo, 2018), authors propose stochastic sub-word tokenization and show that this can improve the robustness of a Machine Translation system. Given we use a pre-trained BERT model with a fixed vocabulary, we consider an adaptation of this method– we first consider an index-based split of each word followed by BERT-based tokenization of each split. Then, we sample each such token with  $\frac{1}{4}$ -th the probability of the original word tokenization. We call this method BERT-based Sub-word Tokenization (B-ST). For example, consider the word ‘fly’, the subwords and the sampling probabilities are as follows.

Manual Splits	BERT sub-word(s)	Sampling Pr
None	‘fly’	0.666
‘f’, ‘ly’	‘f’, ‘ly’	0.167 (0.666/4)
‘fl’, ‘y’	‘f’, ‘l’, ‘y’	0.167 (0.666/4)

**Adversarial Logit Pairing (ALP)** In (Einolghozati et al., 2019), authors proposed the use of data augmentation (data generated via back-translation) along with regularization terms that penalize (via  $L_2$  distance) the model for outputting different intent and slot labeling logits for a training sample and its noise-augmented version. We adapt this approach for our setting, where the A-DA training set is augmented with multiple noise types. For each mini-batch with  $K_I$  pairs of samples derived from an input utterance  $x$  and  $K_S$  pairs of slot values, the loss function looks as follows,

$$\mathcal{L} = \frac{1}{K_I} \sum_{i=0}^{K_I} \|I(x) - I(\tilde{x})\| + \frac{1}{K_S} \sum_{i=0}^{K_S} \|S(x) - S(\tilde{x})\|$$

where  $I$  and  $S$  represents the intent and slot labeling logits respectively.

## 5 Experiments

We first evaluate the trade-off between use of data augmentation during pre-training vs. fine-tuning

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	86.5	99.0	98.7	96.3	91.2
+MLM Aug. <sup>5%</sup>	<b>98.7</b>	97.4	<b>95.6</b>	86.8	98.8	98.4	<b>96.6</b>	91.9
+Train Aug. <sup>20%</sup>	<b>98.7</b>	<b>97.8</b>	95.5	<b>94.6</b>	<b>99.1</b>	<b>98.8</b>	96.4	<b>95.3</b>
+MLM & Train Aug.	98.6	97.6	95.5	94.5	99.0	<b>98.8</b>	96.0	95.1

Table 4: Robustness of BERT with and without augmentation (Aug.) of misspellings at pre-training time (MLM) and IC/SL training time (Train.) to misspelling phenomena in the ATIS and Snips datasets measured by intent classification accuracy (IC Acc.) and slot labeling F1 (SL F1) scores.

time. We focus the majority of our analysis on fine-tuning time augmentation as we show it outperforms pre-training augmentation.

### 5.1 Pre-training vs. Fine-tuning DA for Synthetic Misspellings

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. Our results, shown in Table 4, empirically demonstrate that noisy pre-training does not perform better than training augmentation (esp. on the noisy data). While noisy pre-training generates marginal improvements on SL F1 (avg. +0.3), training time augmentation produces substantially larger gains (+6.1 on SL F1 and +0.5% on IC acc). Due to this finding, we only consider data augmentation at the fine-tuning stage in the experiments that follow.

### 5.2 Results

Table 5 highlights the performance of the various methods (represented by the different rows) for IC accuracy and SL F1 on the original test set (averaged over the control sets for each noise type), treatment sets for the individual noise types, and an average gain over the control and treatment sets of each approach over the BERT IC-SL baseline. Figures in **bold** indicate the best performance for each column.

The diagonal cells under the data augmentation (DA) section, highlighted in yellow, represent the effect of augmentation with a particular noise type and testing on it. While this idea of *Noise-Aligned Data Augmentation (NADA)* has been extensively explored in prior work, we observe that only 5 of the yellow cells (namely, punc. IC, casing SL for ATIS and casing IC, abbrev. SL, misspell SL for Snips) result in the best performing models (out of 26 metrics on noised-test sets). This indicates that

augmentation with other noise types (cross-noise) has beneficial (side-)effects on the robustness to a particular noise type; we term this phenomena as *Cross-noise Robustness Transfer*.

We discuss the results of our experiments in three sections. First, we summarize the performance gains of the different models across all the test sets. Then we analyze the performance on original test sets. Lastly, we analyze performance on the noisy test data.

#### 5.2.1 Overall Performance

**Data-augmentation approaches (DA)** We find that A-DA with BP aggregation outperforms all the approaches on ATIS (IC:+8.4%, SL:+14.4) and SNIPS (IC:+13.1%, SL:+15.6). NADA approaches, while often successful for a particular noise type, do not perform well against all noise types and hence, aggregate DA methods are deemed necessary. Further, we observe that injecting a particular noise in training data provides robustness against other noise types. Thus, we attribute the success of the A-DA BP approach to this cross-noise robustness transfer. We find that uniform sampling of noise types for aggregate data-augmentation (A-DA Uni) is always worse than best-proportion aggregation (A-DA BP), whether used by itself or in conjunction with the regularization approaches. Thus, we only showcase the results of using A-DA BP with regularization approaches in Table 5.

**BERT-based Sub-word Regularization (B-SR)** B-SR, by itself, provides marginal gains for ATIS (IC:+2.7%, SL:+2.3) and degrades the performance of the baseline on SNIPS (IC:−0.8%, SL:−3.3). For ATIS, the gains stem from minor improvements seen on paraphrase and synonym test-data; for SNIPS, we see a consistent degradation in performance for both the original data and all noise types. In contrary, when

Model	IC Accuracy										SL F1									
	orig. <sup>E</sup>	abv.	case	spl.	mor.	para.	punc.	syn.	gain <sup>E</sup>	orig. <sup>E</sup>	abv.	case	spl.	mor.	para.	punc.	syn.	gain <sup>E</sup>		
ATIS	BERT IC/SL	97.9	85.5	72.2	97.4	<b>97.4</b>	77.8	98.9	76.8	0.0	94.5	78.0	22.7	86.5	94.7	89.8	87.8	91.0	0.0	
	+B-SR	95.7	84.8	74.7	97.6	96.2	87.2	98.5	87.8	+2.6	92.8	83.6	38.5	92.7	93.1	87.7	84.1	88.4	+2.3	
	+DA																			
	abbrev. (15%)	97.8	<b>88.2</b>	72.7	97.3	97.1	78.1	<b>99.2</b>	78.1	+0.7	94.1	<b>90.8</b>	36.2	90.0	94.7	89.9	85.7	<b>91.6</b>	+4.0	
	casing (50%)	97.2	84.5	<b>98.2</b>	97.0	95.4	83.6	<b>99.2</b>	82.3	+4.8	93.4	81.1	<b>94.9</b>	88.4	92.9	88.5	85.8	89.8	+9.9	
	misspl. (20%)	97.4	86.2	75.4	<b>97.6</b>	96.5	84.1	<b>99.2</b>	81.7	+2.0	94.5	87.8	50.1	<b>94.6</b>	94.9	89.4	87.1	90.7	+6.3	
	morph. (10%)	97.6	85.5	72.2	97.2	<b>96.2</b>	77.5	<b>99.2</b>	75.7	-0.4	94.6	77.9	18.6	86.9	<b>94.8</b>	89.5	89.1	91.0	-0.4	
	para. (15%)	97.3	85.2	70.8	97.1	<b>96.5</b>	<b>91.4</b>	98.1	<b>91.8</b>	+3.5	93.2	75.6	27.3	85.9	93.9	<b>88.8</b>	93.1	90.1	+0.4	
	punc. (15%)	97.8	86.9	67.9	96.8	96.8	87.5	<b>99.2</b>	85.6	+2.0	93.1	77.2	14.3	85.4	93.8	88.2	<b>93.5</b>	89.8	-1.4	
	syn. (10%)	<b>97.9</b>	88.2	72.4	97.0	96.5	91.0	98.9	<b>90.5</b>	+4.1	<b>94.8</b>	78.3	24.2	87.7	94.8	<b>90.4</b>	89.5	<b>91.1</b>	+0.8	
	+A-DA																			
	Uniform (Uni)	97.3	87.5	89.5	97.5	95.9	89.8	<b>99.2</b>	90.2	+6.1	94.5	90.6	85.3	93.3	<b>95.0</b>	89.6	93.9	90.8	+12.6	
	Best Proportion (BP)	97.5	89.2	<b>98.3</b>	<b>98.0</b>	97.1	<b>91.4</b>	<b>99.2</b>	91.7	<b>+8.3</b>	94.5	<b>92.0</b>	<b>94.9</b>	<b>95.0</b>	<b>95.0</b>	89.4	<b>94.4</b>	<b>90.5</b>	<b>+14.4</b>	
	+A-DA + ALP	97.0	86.5	89.2	97.2	95.1	91.0	98.9	90.2	+5.9	94.2	88.0	84.2	92.8	94.7	89.2	93.3	91.5	+11.9	
	+A-DA + B-SR	97.3	<b>91.6</b>	97.9	97.5	96.5	89.3	<b>99.2</b>	88.4	+7.7	93.8	90.9	94.5	94.6	93.8	89.4	94.0	89.7	+13.7	
+A-DA + ALP + B-SR	96.8	89.2	97.7	97.3	95.9	89.9	98.9	89.3	+7.3	93.5	90.4	94.3	94.4	93.5	89.3	94.0	88.8	+13.3		
SNIPS	BERT IC/SL	98.3	97.6	23.3	98.7	98.0	98.5	—	98.5	0.0	<b>97.0</b>	88.1	5.0	91.2	96.5	94.9	—	<b>96.3</b>	0.0	
	+B-SR	98.2	98.3	20.2	98.4	98.0	96.8	—	98.3	-0.8	93.9	84.6	3.0	92.2	91.6	91.1	—	92.5	-3.3	
	+DA																			
	abbrev. (15%)	98.3	<b>98.3</b>	24.7	98.1	98.3	98.1	—	98.3	+0.2	96.8	<b>96.0</b>	3.0	92.7	<b>97.4</b>	<b>95.2</b>	—	96.1	+1.4	
	casing (50%)	98.6	98.0	<b>98.7</b>	98.2	98.3	98.3	—	98.0	+12.5	96.1	89.6	<b>90.2</b>	91.5	97.2	93.8	—	94.3	+14.0	
	misspl. (20%)	98.7	97.3	24.6	<b>98.9</b>	98.3	98.8	—	98.7	+0.4	96.6	91.3	+5.1	<b>94.9</b>	97.0	95.0	—	95.9	+1.1	
	morph. (10%)	98.3	98.3	19.4	98.4	<b>98.0</b>	98.3	—	98.7	+2.7	96.8	85.6	1.4	91.3	<b>96.8</b>	95.1	—	96.0	-2.7	
	para. (15%)	98.6	97.6	29.1	98.3	98.0	98.5	—	98.2	+0.9	94.3	84.6	9.2	88.1	<b>96.0</b>	<b>92.5</b>	—	93.5	-1.8	
	syn. (10%)	98.3	98.0	21.1	98.5	98.3	98.1	—	<b>98.8</b>	-0.3	96.7	87.5	5.5	91.2	97.2	94.2	—	<b>95.5</b>	-0.2	
	+A-DA																			
	Uniform (Uni)	98.6	97.6	91.2	<b>99.0</b>	98.3	98.8	—	98.3	+11.5	96.4	94.3	67.1	93.9	96.7	94.7	—	95.3	+11.6	
	Best Proportion (BP)	98.8	98.6	98.6	98.9	<b>98.7</b>	<b>99.2</b>	—	98.8	+13.1	95.8	95.7	<b>90.4</b>	94.6	96.6	94.8	—	95.0	<b>+15.6</b>	
	+A-DA + ALP	98.9	98.6	98.6	98.7	<b>98.7</b>	99.0	—	99.2	+13.1	95.8	94.8	86.5	94.6	96.6	94.8	—	94.9	+14.8	
	+A-DA + B-SR	<b>99.2</b>	<b>99.0</b>	<b>99.0</b>	98.5	98.9	98.0	<b>99.2</b>	—	<b>99.3</b>	<b>+13.2</b>	95.2	94.3	89.5	94.0	94.7	93.7	—	94.5	+14.5
	+A-DA + ALP + B-SR	98.8	98.6	98.5	99.0	98.3	99.0	—	<b>99.3</b>	+13.1	95.0	93.5	87.9	94.1	94.5	92.9	—	94.1	+13.8	

Table 5: Intent Classification (IC) and Slot Labeling (SL) metrics for the different approaches on ATIS and SNIPS averaged over 3 seed runs. The column with the Orig.<sup>E</sup> header reports average accuracy over the different phenomena control sets, the red-highlighted columns showcase results on the noised-sets and the Gain<sup>E</sup> shows the average gain over the BERT IC/SL baseline across all datasets. The percentage of augmented data for the (DA) methods is shown in brackets. Yellow cells highlight performance of Noise-Aligned Data Augmentation (NADA) approaches. As SNIPS lacks a punctuation noised-set, the results are omitted for this condition.

coupled with A-DA BP approach, it produces the best performing model on SNIPS IC (+13.2%) marginally edging out A-DA’s gain (+13.1%) and a fraction behind the best performing models in other categories.

**Adversarial Logit Pairing (ALP)** While ALP coupled with A-DA BP boosts performance when compared to the (BERT IC-SL) baseline model on ATIS (IC:+5.9%, SL:+11.9) and SNIPS (IC:+13.1%, SL:+14.8) data, it does not yield the best-performing model robust to all noise types.

An amalgamation of the approaches, although better than the baseline on ATIS (IC:+7.3%, SL:+13.3) and SNIPS (IC:+13.1%, SL:+13.8) data, lacks behind the best-performing models. This indicates that the approaches A-DA, ALP, and B-SR don’t provide complimentary benefits and thus, cannot simply be combined to construct a more robust model.

## 5.2.2 Performance on the Original Test Sets

We observe that synonym-based DA (with 10% synonyms) results in the best performing model

on both IC and SL performance for ATIS. A key reason for this is that ATIS uses domain-specific vocabulary— for example, training data maps utterances with the world ‘flight’ to atis\_flight and ‘plane’ to atis\_aricraft even when the words flight and plane may be used interchangeably in test utterances. Synonym-based DA helps the model to generalize better by allowing the model to avoid this spurious co-relations between words and intents and consider the larger context of the sentence/utterance. For SNIPS, which has a more diverse vocabulary, well distributed over the test and training set compared to ATIS, the original BERT IC-SL performs the best on SL F1, whereas the model A-DA (BP) + B-SR performs the best on IC accuracy. The diverse tokenization of words, caused by both A-DA and B-SR, ensures that words in movie/book/song names, which may often be indistinguishable, does not derail the model’s IC and the model rather pays more attention to the carrier phrases for classifying intent.

### 5.2.3 Performance on Noised Test Sets

For each noise type, we design a control set that contains only the clean version of the noised utterances in the treatment set. While we only list average performance on these original test sets in Table 5, our analysis below often compares the model performance on the treatment to the corresponding control sets (and not the the average metrics). The detailed results are included in the appendix.

**Abbreviations** Abbreviation degrades IC ( $-12.34\%$  for ATIS,  $-0.7\%$  for SNIPS) and SL performance ( $-16.47$  for ATIS,  $-8.87$  for SNIPS). Note that the degradation on ATIS is of higher magnitude than on SNIPS. Upon investigation, we observe that ATIS abbreviations are often specific to the travel domain (e.g., “tix” vs. “tickets”), whereas abbreviations for SNIPS are more general (e.g., “@” vs. “at”). We hypothesize that BERT is more likely to have been trained on the abbreviations present seen in the SNIPS noised set than for ATIS, and therefore more robust on SNIPS.

Augmentation with only abbreviation data boosts performance on SL-F1 but provides small gains on IC accuracy. We find that the coverage of test abbreviations (injected manually) in the augmented training data (injected synthetically) is far lower for ATIS than for SNIPS (40% vs. 74%) and this results in a higher disparity between IC performance on clean (97.86%) vs. noised (88.22%) test-set for ATIS even after augmentation. Further, we observe that augmentation with synonyms also provides a boost in IC accuracy for abbreviation injected test-data for both ATIS and SNIPS. This cross-noise robustness transfer further boosts the accuracy when the model is trained with BP A-DA (that contains 15% abbreviation and 10% synonym data). When further coupled with B-SR, the best performing model achieves an IC accuracy of 91.58% for ATIS ( $+5.06\%$ ) and 99.25% ( $+0.93\%$ ) for SNIPS. For SL, BP performs the best for ATIS while training data augmentation with abbreviation performs the best for SNIPS.

**Casing** IC and SL performances 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 often match character-level sub-word tokens (eg. ‘would’  $\rightarrow$  {‘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 the correct intent classes or the slot-labels. Unfortunately, casing noise is orthogonal to all the other noise types (as injection other noise types does not improve model performance on casing test-set) and thus, does not benefit from cross noise-robustness transfer. While the best performing model for ATIS (IC: 98.28%, SL: 94.88) and SNIPS SL (90.37) is the BP A-DA, it is only marginally better ( $< 0.2$ ) than the training data augmentation approach, which is a better for SNIPS IC. Note that as the BP A-DA approach considers augmented data will all the noise types, it significantly increases the training time per epoch (see appendix for run-times of the various approaches). Hence, if one desires a model to only be robust to casing noise, we suggest using training data augmentation with capitalized words.

**Misspelling** Test-time misspellings do not impact IC accuracy more than 0.2% points; a misspelled word in the utterances changes the sub-token breakdown of that word (‘what’ vs. {‘wa’, ‘t’}) that, in turn, does not change the intent of the sentence. 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 large drop in SL F1 (ATIS:  $-8.6$ , SNIPS:  $-5.1$ ). For ATIS, we noted that slot values for slot labels such at period of day (eg. ‘night’ vs. ‘nite’) are more prone to mis-classification when injected with misspelled noise in comparison to slot values such as day names (‘sunday’ vs. ‘sunday’). Augmentation of training data with 20% misspellings increases SL F1 (ATIS:  $+8$ , SNIPS:  $+3$ ) making it within 1.2 points of performance on the control 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 noise seen at test time. Further, we observe that other noise types such as abbreviation, which also increases the model’s familiarity to sub-word tokens, also boosts the model’s accuracy to test-time misspelling; this cross-noise robustness transfer makes BP A-DA the best performing model for ATIS. Given tokenization is a key issue for the performance drop, we observe B-SR provides a boost in model performance, but comparatively less than A-DA approaches. Also, coupling A-DA + B-SR doesn’t necessarily boost the accuracy further.

**Morphological Variants** In contrast to the lack of robustness seen on various NLP tasks in the presence of morphological noise (Tan et al., 2020b), IC and SL models evaluated on ATIS and SNIPS data are pretty robust to morphological variation at test-time, deviating from test-time performance on clean data by at most  $-1.1$  points. We notice that synonyms and paraphrase noise injected at training time helps in improving the models robustness to morphological variance at test time. Upon using A-DA BP, we observe that this cross-noise robustness transfer boosts the model’s performance beyond the baseline model’s performance on the clean data.

**Synonyms** Synonyms decrease the model’s performance on ATIS (IC:  $-19.8\%$ , SL:  $-2.8$ ), while the impact on SNIPS IC/SL is negligible. We note that the model picks up word to intent correlations (‘fare’  $\rightarrow$  ‘atis\_airfare’, ‘flight(s)’  $\rightarrow$  ‘atis\_flight’, ‘plane(s)’  $\rightarrow$  ‘atis\_aircraft’) and when these words are replaced with a synonym in the test data, the model misclassifies the intent. For example, 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 vs. 9.1 words). While introducing synonyms into the ATIS training data boosts IC ( $+13.7\%$ ) and SL ( $+0.7$ ) performance, the best performing models are the ones augmented with paraphrase data (for ATIS IC) and abbreviation data (for ATIS SL). In this setting, the benefit of cross-noise robustness transfer is more pronounced where other noise types can boost the accuracy more than the particular noise type observed at test time. This is also due to the fact that data augmentation with synonyms generated via word-net + language models does not align as well with human-injected synonym noise seen at test time.


**Paraphrase** Paraphrases lead to a significant drop in ATIS performance (IC:  $-18.6\%$ , SL:  $3.0$ ) and a marginal drop for SNIPS (IC:  $-0.2$ , SL:  $-1.9$ ). Similar to the cases of synonyms, we posit that ATIS is impacted due to (1) the lack of diverse carrier phrases in the training set and (2) a greater degree of dis-similarity between the original and the paraphrased test-sets (evident from the

$0.12$  lower BLEU score compared to SNIPS). While, training augmentation with back-translated paraphrases yields comparable scores to the baseline system, ALP (Einolghozati et al., 2019) coupled with Uni A-DA yields the best IC accuracy for both ATIS and SNIPS. For SL, cross-noise robustness transfer with synonyms (for ATIS) and abbreviation (for SNIPS) aids the model in achieving the best performance.

**Punctuation** As mentioned earlier, SNIPS training data already has (is naturally augmented with) punctuation tokens for a sub-set of utterances, we do not consider punctuation as a noise and hence, do not create a test set for SNIPS. BERT makes models robust to punctuation-based noise on IC accuracy, but we observe a 8.6 point difference on SL F1. This happens due to either the (1) over-prediction of punctuation tokens after a slot value as a slot-value (eg. in an utterance ‘... to Pittsburgh.’ both ‘Pittsburgh’ and ‘.’ are identified as city names) or (2) missing out slot values when a punctuation is present around it (eg. in an utterance ‘... from Columbus, Ohio to ...’, the model fails to recognize Columbus and Ohio as slot values). While augmenting with punctuation data boosts model performance, we also noticed a similar boost in SL F1 when augmenting training data with back-translated paraphrase data because machine translation often automatically introduced punctuation tokens. Hence, when coupled together in the A-DA approaches, we observe the best model performance with BP A-DA on ATIS’ SL F1.

## 6 Conclusion

In this paper, we show that SOTA BERT based IC/SL models are not robust to many real world noise types found in production. We further demonstrate that cross-noise robustness transfer – training on one noise type to improve model robustness on another noise type – yields gains for a number of noise type pairs. Through Aggregate Data Augmentation (A-DA), we leverage cross-noise robustness transfer to improve the model’s average performance by  $+10.8\%$  on IC and  $+15$  points on SL F1. Despite the gains we obtain, our benchmark provides substantial headroom to improve model performance on the abbreviation, synonym, and paraphrase noise types. We hope that our benchmark will support future research in this direction and enable the design of robust IC/SL models for goal-oriented dialogue systems.

**Acknowledgements** We would like to thank the anonymous reviewers at multiple venues and the [Amazon Lex team](#)  for their insightful feedbacks and constructive suggestions.

## References

- Vangie Beal. 2021. [The complete list of 1500+ common text abbreviations & acronyms](#).
- Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. *ICLR*.
- Qian Chen, Zhu Zhuo, and Wen Wang. 2019. [Bert for joint intent classification and slot filling](#).
- Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. [Robust neural machine translation with doubly adversarial inputs](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.
- 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. 2018. [Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces](#). *CoRR*, abs/1805.10190.
- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. [Hotflip: White-box adversarial examples for text classification](#). *arXiv preprint arXiv:1712.06751*.
- Arash Einolghozati, Sonal Gupta, Mrinal Mohit, and Rushin Shah. 2019. [Improving robustness of task oriented dialog systems](#). *arXiv preprint arXiv:1911.05153*.
- Adam Ek, Jean-Philippe Bernardy, and Stergios Chatzikyriakidis. 2020. How does punctuation affect neural models in natural language inference. In *Proceedings of the Probability and Meaning Conference (PaM 2020)*, pages 109–116.
- Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Edouard Hovy. 2021. A survey of data augmentation approaches for nlp. Association for Computational Linguistics.
- Karan Goel, Nazneen Rajani, Jesse Vig, Samson Tan, Jason Wu, Stephan Zheng, Caiming Xiong, Mohit Bansal, and Christopher Ré. 2021. [Robustness gym: Unifying the nlp evaluation landscape](#). *arXiv preprint arXiv:2101.04840*.
- Saša Hasan, Carmen Heger, and Saab Mansour. 2015. [Spelling correction of user search queries through statistical machine translation](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 451–460, Lisbon, Portugal. Association for Computational Linguistics.
- Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. 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*.
- Robin Jia and Percy Liang. 2017. [Adversarial examples for evaluating reading comprehension systems](#). *arXiv preprint arXiv:1707.07328*.
- Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. [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)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.
- Taku Kudo. 2018. [Subword regularization: Improving neural network translation models with multiple subword candidates](#). *arXiv preprint arXiv:1804.10959*.
- Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. [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*, pages 881–893, Valencia, Spain. Association for Computational Linguistics.
- Valentin Malykh, Varvara Logacheva, and Taras Khakhulin. 2018. [Robust word vectors: Context-informed embeddings for noisy texts](#). In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 54–63, Brussels, Belgium. Association for Computational Linguistics.
- Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. [Evaluating robustness to input perturbations for neural machine translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8538–8544, Online. Association for Computational Linguistics.
- Aleksandra Piktus, Necati Bora Edizel, Piotr Bojanowski, Edouard Grave, Rui Ferreira, and Fabrizio Silvestri. 2019. [Misspelling oblivious word embeddings](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3226–3234, Minneapolis, Minnesota. Association for Computational Linguistics.
- Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. [Combating adversarial misspellings with robust word recognition](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy. Association for Computational Linguistics.

- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. [Semantically equivalent adversarial rules for debugging NLP models](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. [Beyond accuracy: Behavioral testing of NLP models with CheckList](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. [Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter](#).
- Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020a. [It’s morphin’ time! Combating linguistic discrimination with inflectional perturbations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2935, Online. Association for Computational Linguistics.
- Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020b. [It’s morphin’ time! combating linguistic discrimination with inflectional perturbations](#). *arXiv preprint arXiv:2005.04364*.
- Ottokar Tilk and Tanel Alumäe. 2016. [Bidirectional recurrent neural network with attention mechanism for punctuation restoration](#). In *Interspeech 2016*.
- Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Thomas Goldstein, and Jingjing Liu. 2020. [Freelb: Enhanced adversarial training for language understanding](#). *ICLR*.

## 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.<sup>5</sup>

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

### 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:

<sup>5</sup>[https://nlp.gluon.ai/model\\_zoo/intent\\_cls\\_slot\\_labeling/index.html](https://nlp.gluon.ai/model_zoo/intent_cls_slot_labeling/index.html)

Dataset	Model Type	≈Time Taken
ATIS	BERT IC-SL	13m
	+ B-SR	17m
	+ Uni A-DA	22m
	+ Uni A-DA + B-SR	27m
	+ Uni A-DA + ALP	37m
	+ Uni A-DA + ALP + B-SR	58m
	+ BP A-DA	57m
	+ BP A-DA + B-SR	65m
	+ BP A-DA + ALP	125m
	+ BP A-DA + ALP + B-SR	207m
SNIPS	BERT IC-SL	23m
	+ B-SR	29m
	+ Uni A-DA	37m
	+ Uni A-DA + B-SR	42m
	+ Uni A-DA + ALP	59m
	+ Uni A-DA + ALP + B-SR	70m
	+ BP A-DA	102m
	+ BP A-DA + B-SR	110m
	+ BP A-DA + ALP	186m
	+ BP A-DA + ALP + B-SR	227m

Table 6: Table showing the time taken to run 30 epochs of a model training + inference on test data. As the test data size varies based on the noise type (since different treatment sets have different number of utterances), all numbers are shown for the abbreviation control set.

```
1 nltk==3.3
2 gluonnlp==0.8.1
3 mxnet==1.3.0
4 numpy==1.14.3
5 scikit-learn==0.23.1
6 scipy==1.1.0
7 torch==1.7.1
8 transformers==4.2.2
9 seqeval==0.0.12
```

Listing 1: requirements.txt

### A.5 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](https://nlp.gluon.ai/model_zoo/intent_cls_slot_labeling/index.html)).

### A.6 Model Training and Inference Times

Training for 40 epochs plus inference on the test set takes approximately 13 minutes on the ATIS dataset and 23 minutes for the SNIPS data-set for both the baseline and training augmentation approaches. The time taken significantly increased for the other

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

Table 7: 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.

model types and are listed in Table 6.

### 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 that 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 7.