Data Augmentation for Voice-Assistant NLU using BERT-based Interchangeable Rephrase

Akhila Yerukola[*], **Mason Bretan**[*], **Hongxia Jin**
Samsung Research America
{a.yerukola, mason.bretan, hongxia.jin}@samsung.com

Abstract

We introduce a data augmentation technique based on byte pair encoding and a BERT-like self-attention model to boost performance on spoken language understanding tasks. We compare and evaluate this method with a range of augmentation techniques encompassing generative models such as VAEs and performance-boosting techniques such as synonym replacement and back-translation. We show our method performs strongly on domain and intent classification tasks for a voice assistant and in a user-study focused on utterance naturalness and semantic similarity.

1 Introduction

With conversational assistants becoming more and more pervasive in everyday life, task-oriented dialogue systems are rapidly evolving. These systems typically consist of Spoken Language Understanding (SLU) models, tasked with determining the domain category and the intent of user utterances at each turn of the conversation.

The ability to quickly train such models to meet changing and evolving user needs is necessary. However, developers often find themselves in situations with access to very little labeled training data. This is especially true when new functions are deployed, and a large user base has not had a chance to utilize the function, thus, limiting the number of available utterances that can be labeled for training. Furthermore, the process of labeling large amounts of data can be time consuming and expensive. More recently, these challenges have been both enhanced and complicated by privacy concerns and legislation that may prevent the use of user utterances for training.

Much of the recent research addressing data paucity has focused on pre-training using self-supervision and vast amounts of unlabeled data (Devlin et al., 2018; Radford et al., 2018, 2019). Pre-trained models can later be fine-tuned with a much smaller amount of labeled data for specific tasks. In this work, instead of pre-training, we explore methods that enhance and expand the task-specific training set by using data augmentation. While models such as BERT prove to be both useful and relevant, we show that data augmentation during the fine-tuning stage can boost performance even on these large pre-trained models. We implement and compare several pre-existing techniques for data augmentation on Natural Language Understanding (NLU) tasks such as domain classification (DC) and intent classification (IC) for a voice assistant. We also introduce a new method of data augmentation called *Interchangeable Rephrase* (IR) with the goal of “rephrasing” an existing utterance using new language while maintaining the original intent or goal (see Table 1).

Utterance	how do I make a margherita pizza
Domain	Recipe
Intent	Return_Recipe
Rephrase	show me how to cook a margherita pizza
Utterance	can you check air quality in santa rosa
Domain	Weather
Intent	Air_Quality
Rephrase	is it possible to check air quality in santa rosa
Utterance	delete all emails which have come from hotel.com
Domain	Email
Intent	Delete_Emails
Rephrase	get rid of all emails which have come from hotel.com

Table 1: An illustrative example of a generated rephrase using our Interchangeable Rephrase, while still maintaining the original domain and intent.

2 Related Work

Recurrent neural network (RNN)-based VAE generative models (Kingma and Welling, 2013; Bowman et al., 2015) explicitly model properties of utterances like topic, style, and other higher-level syntactic features. The variational component helps in generating diverse text, thus, we use VAEs as a candidate for transforming and augmenting text

[*] equal contribution

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1852–1860
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

data in our experiments.

Building upon unconditioned text generation, Conditional VAE (CVAE) (Sohn et al., 2015; Hu et al., 2017) generates more relevant and diverse text conditioned on certain control attributes, e.g. tense, sentiment (Hu et al., 2017), style (Ficler and Goldberg, 2017). In this work, our goal is to maintain semantic similarity, therefore, we generate text by conditioning on the original intent or goal.

Guu et al. generate novel sentences from prototypes by exploiting analogical relationships of sentences. They have shown that generated sentences have a varied style. We extend this idea by using prototype utterances and then editing it into a new utterance or rephrase, using both the VAE and CVAE architecture, to generate augmented rephrase data. We call these models VAE-edit and CVAE-edit.

Back translation (BT) is the process of translating an utterance in a certain language to another language and then translating it back to the original language. Certain question answering (QA) models (Yu et al., 2018) have observed that back-translation generates diverse paraphrases, while preserving the semantics of the original sentences. In our case, we use back-translations as rephrases to augment data.

Another form of simple augmentation techniques is EDA: easy data augmentation (Wei and Zou, 2019). They consist of four operations: synonym replacement, random insertion, random swap and random deletion. They show a boost in text classification tasks with these operations on smaller datasets. Since the number of labeled of spoken utterances are limited, we compare to this approach for our NLU tasks.

Automatic speech recognition (ASR) module, which converts audio input into text, introduces some errors in the process before feeding it into downstream NLU tasks. Without modifying ASR or NLU components, an utterance correction module can be used to help with denoising data (Freitag and Roy, 2018). The reconstructed utterances can be further used as data augmentation.

PPDB is a well known paraphrase database consisting of automatically generated paraphrases (Ganitkevitch et al., 2013). We use this database to rephrase utterances by identifying short phrases within the utterance and replacing them with a related phrase according to the database and POS (e.g. “there is a lot of” → “there are plenty of”).

3 Interchangeable Rephrase

BERT is pre-trained using two tasks: Masked LM (MLM) and Next Sentence Prediction (NSP). In our rephrase task, the end-objective is almost identical to the MLM training, and only this procedure is used to train our self-attention model. MLM training allows the model to predict appropriate word(s) to replace the masked token depending on the context of the rest of the phrase. Each input token corresponds to a final hidden vector that is fed into an output softmax over the vocabulary. Thus, to rephrase an utterance that has more or fewer tokens than the original, the desired number of tokens for the rephrase must be known a priori.

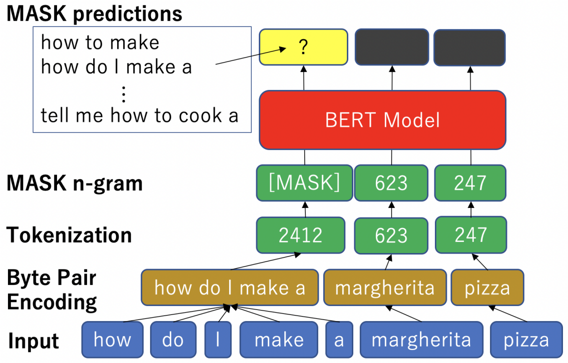


Figure 1: Overview of BERT based interchangeable rephrase (BERT-IR) - BPE is used is to encode n-grams into a single token. Thus, the model’s vocabulary is comprised of tokens representing both single words and sequences of words. The model then computes a softmax over the vocabulary representing the vector of the masked input token allowing for a final output that may be a different word length than the original input.

To allow rephrases with unknown lengths, we use byte pair encoding (BPE) to group word n-grams into single tokens (Sennrich et al., 2015). By using BPE, an individual token may represent a sequence of several words, but the model can still be trained to predict only a single token. We perform BPE on a set of training relevant to the end tasks to get the most frequent n-gram sequences and include these in the model’s vocabulary. Similarly to PPDB, we assume that many of these n-grams are synonyms and are interchangeable. For example, in the context of a virtual assistant skill that enables finding and reciting recipes, n-gram short phrases such as a “how to make”, “tell me how to cook” and “teach me to make” can all be used in the place of the n-gram “how do I make” in the utterance “how do I make a margherita pizza.” This interchangeable property is the foundation of our rephrase and data augmentation system.

The BERT-like self-attention model allows for the predictions to be made based on the context of the input utterance and BPE tokenization allows for variable-length outputs while still only having to predict a single token. Though the new BPE-based vocabulary requires re-training of the model (pre-trained BERT cannot be used), it remains structurally the same as the original BERT model. We refer to our BERT based interchangeable rephrase model as **BERT-IR**. Figure 1 provides an overview of the rephrase model.

In order to maintain intent such that the bigram "turn on" in the utterance "turn on the lights" is not inadvertently replaced with "turn off", negative examples and an intent feature are included in a fine-tuning step. The fine-tuning process minimizes a loss function based on the cosine similarity function

$$sim(\tilde{X}, \tilde{Y}) = \frac{\tilde{X}^T \cdot \tilde{Y}}{|\tilde{X}||\tilde{Y}|} \tag{1}$$

where $\vec{X}$ and $\vec{Y}$ are two equal length vectors and negative examples are included through a softmax function

$$P(\tilde{R}|\tilde{Q}) = \frac{\exp(sim(\tilde{Q}, \tilde{R}))}{\sum_{\tilde{d} \epsilon D} \exp(sim(\tilde{Q}, \tilde{d}))} \tag{2}$$

where $\vec{Q}$ is a one hot ground truth token label derived from the input Q, and $\vec{R}$ represents the output vector of Q. D is the set of three vectors that includes $\vec{R}$ and two vectors derived from two negative examples in the training set. The network then minimizes the following differentiable loss function using gradient descent

$$-log \prod_{(Q,R)} P(\tilde{R}|\tilde{Q}) \tag{3}$$

We emphasize that it is not necessary to use a BERT model for this rephrase method. It is technically possible to use a model architecture identical to that of BERT. However, given that this is not a fine-tuning task and the model needs to be fully re-trained to support the new n-gram tokens a less resource intensive and data hungry model is preferred. In our experiments we use a BERT-like model that leverages self-attention, but has only a fraction of the total parameters of BERT.

4 Experiments

In the following experiments, we examine and compare our proposed method with various data augmentation techniques in the context of utterance generation (rephrase) for voice assistants.

First, we study the properties of the utterances the systems are capable of generating. An ideal data augmentation method should create data that either expands upon or fills in missing gaps of the original training distribution, while still being inherently natural and meaningful. In this context, given an input utterance (from the original distribution), the goal is to generate rephrases that are semantically similar, yet, different enough to positively alter the original training distribution. This is measured in our next two experiments in which we compare the performance of augmented datasets by training an utterance domain classifier (DC) and an intent classifier (IC).

Finally, we perform a user study to examine the quality of utterance generation in terms of naturalness and semantic similarity.

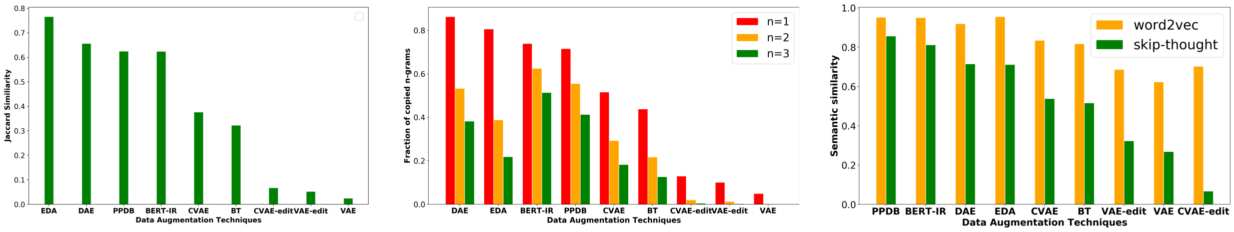
Data. We use an original dataset comprised of utterances for a set of 63 skills (domains) for a voice assistant. The skills range from playing a song on Spotify to turning on/off in-home appliances to providing the weather. The developer of each skill provides a set of training utterances that users can say as an entry point to the function. Each utterance is annotated with an intent, and on average there are 530 utterances per skill (maximum 2000 and minimum 9).

For the DC task, of the 43540 utterances in the dataset, 6590 are held out for validation and the remaining utterances are used for training and processed for augmentation. Additionally, we use a separate test set that was collected and labeled through user trials and crowd sourcing. Each domain in this test set has roughly 900 test utterances.

For the IC task, we consider 5 different skills (domains) from the above dataset: Weather, SystemApp, SmartThings, TvControl and TvSettings. We allocate 30% of the data in each skill for testing and use the remaining for training including augmentation. (see Appendix A.3 for more details).

4.1 Experimental Settings

Our BERT-based interchangeable rephrase model uses an architecture that is similar in size and structure to DistilBERT (Sanh et al., 2019). It is possible to use alternative types of models, but we are motivated by the power of self-attention mechanisms for sequential tasks. Our model is pre-trained with roughly 500k utterances from user data (in



(a) **Jaccard Similarity:** ↓ lower is better

(b) **Copied n-gram fraction:** ↓ lower is better

(c) **Semantic Similarity:** ↑ higher is better

Figure 2: Evaluation of automatic metrics: Averaged across all the generated rephrases for each model.

USA) for the voice assistant. The vocabulary is established using a combination of the developer training data and the usage data. After the byte-pair-encoding process, we prune the resulting pairs so that only pairs occurring more than 100 times are used in the final vocabulary. All unigrams appearing two or more times are also included in the vocabulary.

Baselines. Our first baseline is a classifier that is finetuned solely on the original training data with no augmentation. We compare with existing work on data augmentation via VAE/CVAE, VAE/CVAE-edit, back-translation, denoising autoencoder (DAE), easy data augmentation (EDA) and PPDB. We refer the reader to Appendix A.1 for experimental details of the above augmentation techniques.

4.2 Analysis and Comparison of Methods

We apply several automated linguistic metrics to examine differences in quality of generated rephrases. We evaluate the generated rephrases based on how *related* they are to the original utterances. A model which minimizes the word-level overlap (i.e more variation) and increases the semantic similarity the most is presumably ideal.

Jaccard similarity (Jaccard, 1912; Roemmele et al., 2017) is used to measure the proportion of overlapping words between the rephrase and the original utterance. Additionally, for $n = 1, 2, 3$, we measure the proportion of generated n-grams that also appear in the original utterance i.e amount copied from original utterance (See et al., 2019).

We measure semantic similarity at the word-level and sentence-level. We compute the the mean cosine similarity of the word2vec vectors of all pairs of words between a rephrase and the original utterance. We also measure the cosine similarity of the sentence encodings, generated by the skip-thought model (Kiros et al., 2015), of the rephrase and the original utterance (see Figure 2c). The skip-thought model maps sentences sharing semantic and syntactic properties to similar vector representations.

MODEL	HELD OUT	TEST ACC
NO - AUGMENTATION	0.9229	0.8251
VAE	0.9212	0.8379
CVAE	0.9277	0.8447
VAE-EDIT	0.84	0.6891
CVAE-EDIT	0.8617	0.7469
BACK-TRANSLATION	0.9361	0.8881
DAE	0.9307	0.8765
EDA	0.9896	0.8921
PPDB	0.9341	0.8764
BERT-IR	0.9738	**0.9062**

Table 2: Domain classification results.

We average all these metrics across all the generated rephrases for each model (see Figure 2).

4.3 Domain and Intent Classification

The data generated from each augmentation technique is used to train a domain classifier and an intent classifier. Ten classifiers are trained for each task, using the same distribution of training and held out data described previously. Though several of the augmentation techniques are capable of generating more than one utterance per input, here we generate augmented data in a 1-1 fashion i.e, for each original utterance in the training set, a single rephrase is generated (see Figure 3).

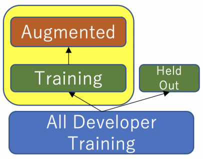


Figure 3: Training distribution for domain and intent classification. All data within yellow boundary box is used for training the classifier.

For both the tasks, we train a classifier on top of the base uncased DistilBERT [1] model (see A.2).

[1] https://huggingface.co/transformers/model_doc/distilbert.html

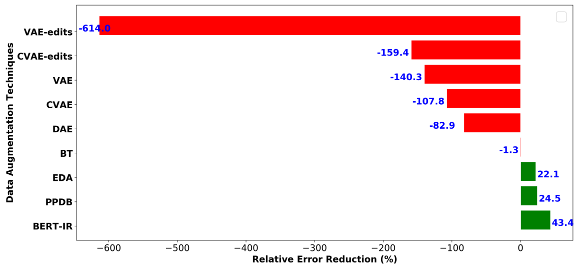


Figure 4: Intent Classification: Relative error reduction of each augmentation type compared to the original “no augmentation” classifier. A positive value (green bar) indicates better performance.

SOURCE	AVG. NATURALNESS	INTENT ACCURACY
HUMAN	3.9	84.5
PPDB	3.1	72.3
OURS	3.4	78.6

Table 3: User study results. The average likert naturalness (where 5 is very natural and 1 is very unnatural) and intent matching accuracy are reported for each of the three utterance sources.

The results on DC task are shown in Table 2. In Figure 4, we show the relative error reduction of each augmentation technique achieved on IC, averaged across all 5 skills (see Appendix A.4 for complete details).

4.4 User study

We performed an online user study where participants completed two tasks. In the first task, participants answered a three item Likert survey regarding the naturalness of the utterance (see Appendix A.5). We compared results for utterances that came from the original developer training (i.e. human-generated), our proposed BERT-IR, and a baseline rephrase using PPDB.

In the second task, users were given a reference utterance and four candidate utterances. Of the four candidates, one of them had an identical intent as the reference, but with different wording, and the other three were random utterances with different intents. The participants were asked to select the candidate with the same intent as the reference. The “correct” candidate was generated from one of the three sources: human, our rephrase, or PPDB rephrase. There were 88 participants, and each participant completed each task 30 times. Results for both tasks are shown in Table 3. Using a 3-way ANOVA, we found significant differences between all three methods on a $p < .05$ level.

5 Discussion

In our experiments, we show that VAE models are capable of generating diverse rephrases. However, these rephrases do not preserve the original meaning. This likely contributed to poorer performance on DC and IC. The discriminator of CVAE models trained to condition on a domain (DC) or intent (IC) helps improve semantic similarity resulting in slightly improved performance of NLU tasks. The VAE-edit and CVAE-edit models perform quite poorly on all comparisons as they don’t necessarily preserve the meaning of the utterance when transforming them into an altered style.

Back-translation yields relatively diverse rephrases; however, it poorly conserves the original meaning. DAE just tends to copy a higher fraction of n-grams, while not changing the meaning of the utterance. EDA has boosted the DC and IC performance, compared to other methods. However, it merely changes a small percent of words in utterances by replacement operations, which are not always grammatically sound.

Our BERT-IR yields the best performance on the NLU tasks, with a relative error reduction of 46.26% on DC and 43.4% on IC as compared to no additional augmentation. Similar to PPDB, our approach generates rephrases with a lower word-overlap and a significantly higher semantic similarity with the original utterance. The user study reveals that BERT-rephrase is an improvement over the PPDB baseline, but does not perform at the level of human generated utterances in terms of naturalness and intelligibility.

For examples of generated rephrases see Table 8 in the Appendix.

6 Conclusion

We introduced BERT-IR, a simple augmentation strategy based on byte pair encoding and a BERT-like self-attention model to generate diverse, natural and meaningful rephrases in the context of utterance generation for voice assistants. We demonstrate that BERT-IR performs strongly on spoken understanding tasks like domain classification and intent classification and in a user-study focused on evaluating quality of rephrases based on naturalness and interpretability (intent preservation).

References

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633*.

Markus Freitag and Scott Roy. 2018. Unsupervised natural language generation with denoising autoencoders. *arXiv preprint arXiv:1804.07899*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org.

Paul Jaccard. 1912. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50.

Diederik P Kingma and Max Welling. 2013. Autoencoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Melissa Roemmele, Andrew S Gordon, and Reid Swanson. 2017. Evaluating story generation systems using automated linguistic analyses. In *SIGKDD 2017 Workshop on Machine Learning for Creativity*, pages 13–17.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. 2019. Do massively pretrained language models make better storytellers? *arXiv preprint arXiv:1909.10705*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491.

Jason W Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

A Appendices

A.1 Implementation and Training Details of Augmentation Strategies

VAE: The decoder and encoder of the VAE model are set as single-layer GRU RNNs with input/hidden dimension of 100/150 and max sample length of 15. Larger input/hidden dimension models performed similarly. To avoid vanishingly small KL term in the VAE module, we use a KL term weight linearly annealing from 0 to 0.15 during training (similar to training procedure in Hu et al.). Training a larger VAE model yielded comparable results.

CVAE: The discriminator is set as ConvNets similar to Hu et al.. Balancing parameters are set to $\lambda_c = \lambda_z = \lambda_u = 0.1$. Training procedure is exactly as Hu et al..

VAE edit, CVAE-edit: We match tuples of utterances which have the same domain and intent. These pairs are used as the prototypes and new utterances during training of VAE-edit and CVAE-edit (similar to training procedure in Guu et al. by constructing edit vectors and concatenating it with z before feeding into the decoder).

Denoising Auto-encoder: The decoder and encoder of the denoising auto-encoder are set as single-layer GRU RNNs with input/hidden dimension of 300/512 with Luong attention. We follow a similar utterance corruption as Freitag and Roy by dropping words randomly whose frequency was greater than 100 (i.e non-content words like ”the”), followed by shuffling of its bigrams while not splitting bigrams that also exist in the original utterance. We train a DAE using this corrupted data. We use the corrected utterance/rephrases generated as augmentation data.

We train the above models until convergence on 2 NVIDIA Tesla V100 GPUs.

Back-translation: We use the open-sourced back-translation system from (Xie et al., 2019). Specifically, we use pre-trained WMT'14 English-French translation models (in both directions) to perform back-translation on each utterance.

EDA: We followed three strategies by Wei and Zou: random insertion, random replacement and random swap.

PPDB: The PPDB [2] database consists of 1 paraphrase rule per line. A standard format of a line is:

```
 LHS ||| PHRASE ||| PARAPHRASE |||
(FEATURE=VALUE )* ||| ALIGNMENT |||
ENTAILMENT
```

If a PHRASE exists in the utterance, we replace it with the PARAPHRASE. Since it's a 1-1 data augmentation, we sample a rephrase from the list of all possible generated rephrases. We use the English-Phrasal PPDB database .

Refer to Table 4 for the average run time for the different augmentations techniques.

Model	Time
VAE (GPU)	1 minute
CVAE (GPU)	1.5 minutes
VAE-edit (GPU)	2 minutes
CVAE-edit (GPU)	3 minutes
Back-translation (GPU)	30 minutes
DAE (GPU)	1 minute
EDA (CPU)	3 minutes
PPDB (CPU)	11 minutes
BERT-Rephrase (GPU)	7 minutes

Table 4: Average time to run inference for augmentation over the training set (36,950 utterances) assuming one generated output per utterance.

A.2 Domain Classification and Intent Classification

For domain classification and intent classification, we train a classifier on top of the base uncased DistilBERT model. We use a maximum sequence length of 128, a dropout rate of 0.1, and a learning rate of 2e-5. We train the classifier for 15 epochs and batch size of 32 on a single NVIDIA Tesla V100.

Splits	Num. of utterances
Train	36950
Held Out	6590
Test	56820

Table 5: Domain Classification Data Statistics

A.3 Training Data Statistics

Domain Classification: Table 5 shows the counts for each training, development (held out), and test set. The partition used for training is also the partition that is augmented (and subsequently also used for training in domain classification, see Figure 3). Each utterance has an average of 6 words. There are 63 different domains in our dataset like music, calendar, calculator, weather etc. We evaluate the automatic metrics from Section 4.2 on the same data used for training the domain classifier.

[2] http://paraphrase.org/#/download

DOMAIN/SKILL	NUM. OF INTENTS	TRAIN DATA SIZE	TEST DATA SIZE
WEATHER	22	732	313
SYSTEMAPP	29	665	286
SMARTTHINGS	61	1190	511
TVCONTROL	19	392	168
TVSETTINGS	40	834	357

Table 6: Intent Classification Data Statistics

Intent Classification: Table 6 shows the training split counts for each of the 5 different domains. Similar to domain classification, the partition used for training is also the partition that is augmented (and subsequently also used for training in intent classification, see Figure 3).

A.4 Intent Classification Results

As shown in Table 6, the amount of training data available to train an intent classifier for each domain is very less. This can explain the poor performance of VAE models and its variations on IC since they require a lot more training data to perform well. This is also illustrated in the results of domain classification where these VAE based models achieve a slight improvement since there is a higher amount of training data available. Please refer to Table 7 for the fine-grained performance results of the augmentation techniques on the 5 domains.

A.5 User Study Details

Participants responded to three items on a Likert-scale of one to five:

1. This utterance is natural. (disagree to agree)
2. It is easy to understand the intent. (disagree to agree)
3. A person is likely to say this to a voice assistant to achieve the underlying goal. (disagree to agree)

MODEL	WEATHER	SYSTEMAPP	SMARTTHINGS	TVCONTROL	TVSETTINGS
NO - AUGMENTATION	0.97129	0.97894	0.95882	0.92814	0.92977
VAE	0.92971	0.91929	0.92745	0.88023	0.88764
CVAE	0.93062	0.95087	0.92941	0.88023	0.88764
VAE-EDIT	0.72966	0.81403	0.68823	0.80239	0.9382
CVAE-EDIT	0.92344	0.9614	0.87058	0.88622	0.87921
BACK-TRANSLATION	0.97368	0.97192	0.95294	0.93413	0.95505
DAE	0.94976	0.95438	0.92352	0.8982	0.91853
EDA	0.99361	0.98245	0.96078	0.93413	0.92696
PPDB	0.98564	0.98245	0.9666	0.93413	0.93539
BERT-IR	**0.9968**	**0.98596**	**0.97058**	**0.94011**	**0.94382**

Table 7: Test accuracy for Intent Classification

Model	Generated Rephrase
Original	*barbershop near me*
VAE	send an email to tony
CVAE	barbershop near me
VAE-edit	is mike's bikes closed on weekends?
CVAE-edit	where is the highest rated dentist nearby?
Back-translation	the ”Teller” Argile been in close commute for us!
DAE	barbershop near near me
EDA	near me barbershop
PPDB	barbershop near me
BERT-IR	barbershop close to my current location

Model	Generated Rephrase
Original	*call an uber espanol to los altos taqueria for me*
VAE	what is the time difference between kuala lumpur and and 40 feet
CVAE	call holly davis espanol to a stadium in brooklyn
VAE-edit	order a lyft premier to 1
CVAE-edit	order a lyft lux for 2 to a cake for me
Back-translation	call a espanol uber at altos taqueria commands for me
DAE	look for an espanol espanol los altos
EDA	call an uber espanol to los for me altos taqueria
PPDB	'm calling the uber espanol to los altos taqueria for me
BERT-IR	please contact an uber espanol to los altos taqueria for me

Model	Generated Rephrase
Original	*how is today's visibility?*
VAE	send a lyft premier to see you
CVAE	how is today's visibility?
VAE-edit	what's the weather going to be here
CVAE-edit	navigate to golden gate bridge
Back-translation	How do we make today's visibility?
DAE	what is today's visibility?
EDA	how is nowadays's visibility?
PPDB	how is today's visibility?
BERT-IR	i want to know how's today's visibility

Table 8: Examples of generated rephrases by the different augmentations techniques