Simple is Better! Lightweight Data Augmentation for Low Resource Slot Filling and Intent Classification

Samuel Louvan
University of Trento
Fondazione Bruno Kessler
slouvan@fbk.eu

Bernardo Magnini
Fondazione Bruno Kessler
magnini@fbk.eu

Abstract

Neural-based models have achieved outstanding performance on slot filling and intent classification, when fairly large in-domain training data are available. However, as new domains are frequently added, creating sizeable data is expensive. We show that *lightweight augmentation*, a set of augmentation methods involving word span and sentence level operations, alleviates data scarcity problems. Our experiments on limited data settings show that lightweight augmentation yields significant performance improvement on slot filling on the ATIS and SNIPS datasets, and achieves competitive performance with respect to more complex, state-of-the-art, augmentation approaches. Furthermore, lightweight augmentation is also beneficial when combined with pre-trained LM-based models, as it improves BERT-based joint intent and slot filling models.

1 Introduction

In task-oriented dialogue systems, a spoken language understanding component is responsible for parsing the user utterance into a semantic representation. This is often modeled as a semantic frame (Tur and De Mori, 2011), and typically involves *slot filling* and *intent classification*. For example, in the utterance "*book in Southern Shores for 8 at Ariston Cafe*", the intent is BOOKING A RESTAURANT and the corresponding slot values and slot names are "*Southern Shores*" (CITY_NAME), "*8*" (NUMBER_OF_PEOPLE), and "*Ariston Cafe*" (RESTAURANT_NAME).

Although neural-based models (Qin et al., 2019; Goo et al., 2018; Mesnil et al., 2015) have achieved stellar performance in slot filling (SF) and intent classification (IC), their performance depend on the availability of large labeled datasets. Consequently, they suffer in *data scarcity* situations, which regularly happen when new domains are added to the system to support new functionalities.

One of the methods proposed to alleviate data scarcity is *data augmentation* (DA), which aims to automatically increase the size of the training data by applying data transformations, ranging from simple word substitution to sentence generation. Recently, DA has shown promising potential for several NLP tasks, including text classification (Wei and Zou, 2019; Wang and Yang, 2015), parsing (Sahin and Steedman, 2018; Vania et al., 2019), and machine translation (Fadaee et al., 2017). As for SF and IC, DA approaches typically generate synthetic utterances by leveraging Seq2Seq (Hou et al., 2018; Zhao et al., 2019; Kurata et al., 2016), Conditional VAE (Yoo et al., 2019), or pre-trained Natural Language Generation (NLG) models (Peng et al., 2020). Such approaches make use of in-domain data, and are relatively *heavyweight*, as they require training neural models, which may involve several phases to generate, filter, and rank the produced augmented data, thus requiring more computation time. It is also relatively challenging for deep learning-based models to generate semantically preserving synthetic utterances in limited data settings.

In this paper, we show that *lightweight augmentation*, a set of simple DA methods that produce utterance variations, is very effective for SF and IC

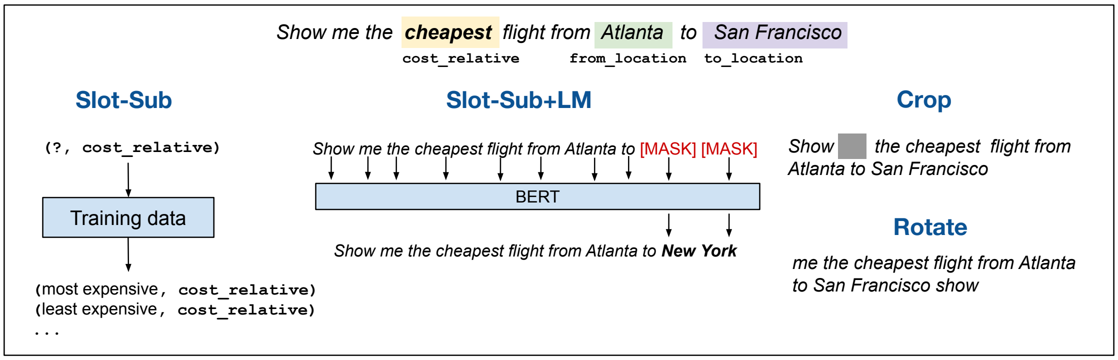

Show me the cheapest flight from Atlanta to San Francisco
cost_relative
from_location
to_location
Slot-Sub
(?, cost_relative)
Training data
(most expensive, cost_relative)
(least expensive, cost_relative)
...
Slot-Sub+LM
Show me the cheapest flight from Atlanta to [MASK] [MASK]
BERT
Show me the cheapest flight from Atlanta to New York
Crop
Show ■ the cheapest flight from Atlanta to San Francisco
Rotate
me the cheapest flight from Atlanta to San Francisco show

Figure 1: Examples of applying *lightweight augmentation* on an utterance in the ATIS dataset.

in a low-resource setting. Lightweight augmentation considers both *text span* and *sentence* variations. The span-level augmentation aims to diversify slot values in a particular text span through a *semantically preserving* substitution of slot values. The sentence-level augmentation seeks to produce alternative sentence structure through crop and rotate (Sahin and Steedman, 2018) operations based on a dependency parse structure.

We investigate the effect of lightweight augmentation both on typical biLSTM-based joint SF and IC models, and on large pre-trained LM transformers based models, in both cases with a limited data setting. Our contributions are as follows:

- We present a lightweight text span and sentence level augmentation for SF and IC. We show that, despite its simplicity, lightweight augmentation is competitive with more complex, deep learning-based, augmentation.
- We show that big self-supervised models, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2020) can perform well under a low data regime, and still benefit from lightweight augmentation.
- The combination of our span based augmentation and transfer learning (e.g. BERT fine-tuning) yields the best performance for most cases.

2 Lightweight Data Augmentation

Given the original training data $\mathcal{D}$, DA aims to generate additional training data $\mathcal{D}'$. For each sentence S in $\mathcal{D}$, an augmentation operation is applied N times, which can be empirically determined. Each augmented sentence S' is added to $\mathcal{D}'$, and the union of $\mathcal{D}$ and $\mathcal{D}'$ is then used to train the model for SF and IC. We describe the lightweight DA operations in the following subsections.

2.1 Slot Substitution (Slot-Sub)

Our first lightweight method, slot substitution, is similar to Gulordava et al. (2018), which is based on substituting a token in a sentence with another token with a consistent syntactic annotation (i.e., part-of-speech or morphology tags). However, unlike Gulordava et al. (2018), our method is not limited to single tokens. As slot filling is a *semantic* task, rather than syntactic, we can naturally extend the method from single tokens (i.e., slot names composed by a single token) to multiple tokens (i.e., slot names composed by multiple tokens, or *spans*[1]), still preserving the semantics associated to a certain slot.

Practically, for slot substitution we take advantage of the fact that SF training data are typically annotated with the BIO format[2]. We exploit the fact that two text spans in different utterances in $\mathcal{D}$ are likely to be semantically similar if they share the same slot label. We randomly pick one span in the S and then perform the substitution (Figure 1 *Left*). For instance, we can substitute the span "*cheapest*", with other spans having the same slot label (i.e.,

[1] We define a span as a sequence of one or more tokens that convey a slot value.

[2] B indicates the beginning of the span, I indicates the inside of the span. O indicates that a token does not belong to any slot. For example, "San Francisco" will be annotated as B-to_location I-to_location.

COST_RELATIVE), such as "*least*" or "*most expensive*".

More formally, we denote a span sp in a sentence S as a slot-value pair $sp = (y, val)$, and we aim to produce an alternative pair $sp' = (y', val')$ such that the slot values are different ($val \neq val'$) and the slot labels are the same ($y = y'$) for both slot-value pairs. To obtain sp', we collect a set of candidates $\mathcal{SP}' = \{sp'_1, sp'_2, ..., sp'_n\}$, by looking for slot spans in other sentences in $\mathcal{D}$ that satisfy our criteria. After that, we randomly sample a span from $\mathcal{SP}^{'}$ to obtain a $sp^{'}$. We replace sp in S with sp' to produce the new augmented sentence $S^{'}$. For example, in the utterance "*show me the cheapest flight from Atlanta to San Francisco*", one of the spans that can be substituted is $sp = (\text{COST\_RELATIVE}, \text{“}\textit{cheapest}\text{”})$. Assuming that from $\mathcal{D}$ we can obtain $\mathcal{SP}^{aug} = \{(\text{COST\_RELATIVE}, \text{“}\textit{least expensive}\text{”}), (\text{COST\_RELATIVE}, \text{“}\textit{most expensive}\text{”}), \dots\}$, we then sample a sp' from $\mathcal{SP}'$ and replace sp in S with sp' to produce S'. Notice that the slot values in sp' are not necessary synonyms of the original slot value, although their slot label must be the same to preserve semantic compatibility.

2.2 Slot Substitution with Language Model (SLOT-SUB-LM)

Our second lightweight method, SLOT-SUB-LM, shares the goal with SLOT-SUB, i.e., to substitute sp with sp'. However, we do not use $\mathcal{D}$ to look for substitute candidates, instead we use a large pre-trained language model to generate the slot value candidates, using the *fill-in-the-blank* style (Donahue et al., 2020). The expectation is that large pre-trained LMs, being trained on massive amount of data, can produce a sensible text span given a particular sentence context, and possibly produce slot values that do not occur in $\mathcal{D}$. While we use BERT for our purpose, virtually any pre-trained LM can be used for SLOT-SUB-LM. Existing works on DA using LMs (Kobayashi, 2018; Kumar et al., 2020) are applied on text classification to replace random tokens in the text, which is not directly applicable to SF. Our approach focuses on *spans* conveying slot values, and include a filtering mechanism to reject retrieved slot spans that are not semantically compatible.

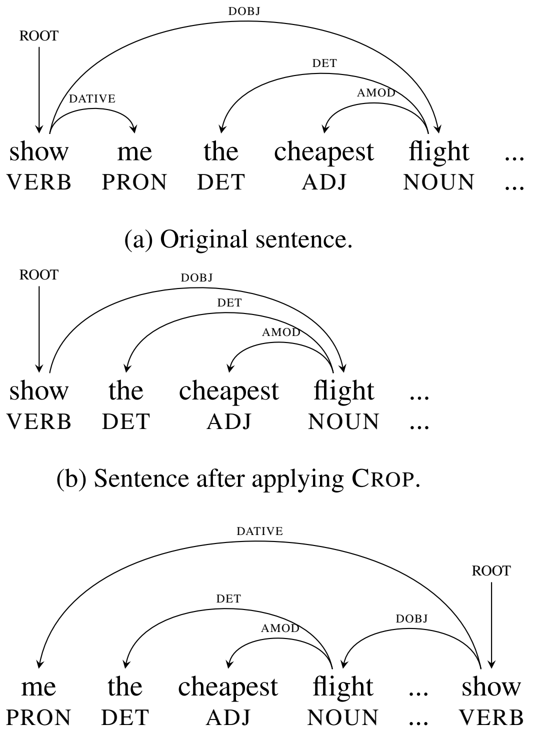
Generating New Slot Values. Given an utterance consisting of one or more slot value spans, we "blank" one of the span and then let the LM to predict the new tokens in the span. For instance, we give "*show me the _____ round trip flight from Atlanta to Denver*" to the LM for blank prediction. Practically, blank tokens are encoded as special `[MASK]` tokens[3] to let the pre-trained LM performing prediction. The decoding of the new tokens is carried out iteratively from left to right (Figure 1 *Middle*) and, to produce the surface form of a token, we apply nucleus sampling (Holtzman et al., 2020) using the top-*p* portion of the probability mass. Nucleus sampling has been empirically shown to be better than beam search, and top-*k* sampling (Fan et al., 2018) to produce fluent and diverse texts.

Filtering. While pre-trained LMs are expected to generate sensible replacements for a span in the utterance, a possible issue is that the new slot span is not semantically consistent with the original one. For example, for the original span "*cheapest*" in "*show me the cheapest round trip flight from Atlanta to Denver*", the LM could output "*earliest*" as a substitution, which does not fit the slot label COST_RELATIVE. To mitigate this issue, we use a binary sentence classifier as a *filter* (**SLOT-SUB-LM+Filter**) to decide whether S and S' are semantically compatible, based on the change made on the slot span. The training of the classifier is composed of a pair S and S', with its binary decision label (i.e., accept or reject S'). To construct the training data, for positive examples (*accept*) we take advantage of the sentence pair produced by SLOT-SUB, while for the negative examples (*reject*) we sample sp' in $\mathcal{D}$ where $y \neq y'$ and replace sp in S with sp' to produce S'. We use the BERT model as the sentence pair classifier and we encode the tokens, w, in both S and S' sentence pairs, as $\texttt{[CLS]}\, w^S_1 w^S_2 \dots w^S_n$ $\texttt{[SEP]}\, w^{S'}_1 w^{S'}_2 \dots w^{S'}_m$. On top of BERT, we add a feed-forward layer that uses the hidden state of the sentence representation, $h_{\texttt{[CLS]}}$, for prediction.

[3] We set the number of masked tokens to be the same as the tokens of the original slot value, e.g. `san francisco` is masked as `[MASK] [MASK]`, although this number could actually be sampled as well.

2.3 CROP and ROTATE

The third lightweight method that we present augments an utterance by changing its syntactic structure. We adopt the augmentation approach from Sahin and Steedman (2018) (Figure 1 *Right*), which is based on two operations, CROPand *rotate*, applied to the dependency parse tree of a sentence. To our knowledge, this approach has not yet been applied to slot filling and intent classification, which is a contribution of our work. *Crop* focuses on particular fragments of a sentence (e.g., predicate and its subject, or predicate and its object), and removes the rest of the fragments, including its sub-tree, to create a smaller sentence. *Rotate* aims to rotate the target fragment of a sentence around the root of the dependency parse structure, producing a new utterance. For example, in the utterance "*show me the cheapest flight from Atlanta to San Francisco*", the word *"me"* can be cropped as it is one of the children of the *root* verb "*show*". While for rotation, the direct object ("*flight*") and its children ("*the cheapest*") are rotated around the root verb. Figure 2 illustrates the relevant dependency structure manipulation.


ROOT
DOBJ
DATIVE
DET
AMOD
show me the cheapest flight ...
VERB PRON DET ADJ NOUN ...
ROOT
DOBJ
DET
AMOD
show the cheapest flight ...
VERB DET ADJ NOUN ...
DATIVE
ROOT
DET
AMOD
DOBJ
me the cheapest flight ... show
PRON DET ADJ NOUN ... VERB

(a) Original sentence.

(b) Sentence after applying CROP.

(c) Sentence after applying ROTATE.

Figure 2: Examples of dependency tree operations on a sentence.

3 Experiments and Results

We experimented our lightweight augmentation approach on three well-known datasets for SF and IC, namely ATIS (Hemphill et al., 1990), SNIPS (Coucke et al., 2018) and FB (Schuster et al., 2018). All datasets are in English. ATIS contains utterances related to flight domain (e.g., searching flight, booking). SNIPS includes multi-domain utterances such as weather, movie, restaurant, etc. FB contains utterances from 3 domains, weather, alarm, and reminder. To simulate the *data scarcity* setting, we follow previous works (Hou et al., 2018; Yoo et al., 2019) and only use *medium*-size (i.e., 1/10) of training data for each dataset. Statistics on the three datasets are reported in Table 1.

As for evaluation, we use standard evaluation metrics, namely the F1-score for SF and accuracy for IC[4]. Performance are calculated as the average score of ten different runs. In order to compare our methods, we use two baselines for slot filling and intent detection: a simple BiLSTM-CRF model, and a state of the art BERT-based model, which is fine-tuned to SF and IC[5]. Each model is trained for 30 epochs, and we apply early stopping criteria.

For both slot substitution (SLOT-SUB) and slot substitution with language model (SLOT-SUB-LM) augmentation methods, we tune the number of augmented sentences per utterance, N, on the dev set of each dataset. For crop and rotate, we use the default parameters from Sahin and Steedman (2018). To produce the dependency parse structure for the utterances in our datasets, we use Spacy[6]. All hyperparameters are tuned on the dev set. More details on the settings is provided in Appendix A. For training the binary classifier for SLOT-SUB-LM+Filter, we generate the same number of positive (*accept*) and negative (*reject*) training instances[7].

In order to allow comparison with more complex data augmentation approaches, we also report

[4] Metric is computed using the standard evaluation script https://www.clips.uantwerpen.be/conll2000/

[5] We use the `bert-base-uncased` model

[6] https://spacy.io/

[7] Details in Appendix B

	Label		**#Utterances ($\mathcal{D}$)**			**#Augmented Training Utterances ($\mathcal{D}'$)**			
Dataset	**#slot**	**#intent**	**#train**	**#dev**	**#test**	**SLOT-SUB**	**SLOT-SUB-LM**	**CROP**	**ROTATE**
ATIS	79	18	0.4K	500	893	3.9K	0.8K	0.8K	1.1K
SNIPS	39	7	1.3K	700	700	6.3K	2.5K	2.6K	3.7K
FB	16	12	3K	4.1K	8.6K	5.4K	5.4K	5.9K	8.5K

Table 1: Statistics of both the original training data $\mathcal{D}$ and the augmented data $\mathcal{D}'$. #train denotes our medium-size training data setup (10% of full training data). $\mathcal{D}'$ is produced by each augmentation method, where the number N of augmentations per sentence is tuned on the dev set.

results obtained with state of the art approaches based on Seq2Seq (Hou et al., 2018) and Conditional Variational Auto Encoder (CVAE) (Yoo et al., 2019). Our implementation is based on the Huggingface library (Wolf et al., 2019), available at `https://github.com/slouvan/saug`.

3.1 Results

Table 2 reports the results on the test sets used in our experiments. As for comparison, we include best-reported scores from two state of the art augmentation methods, namely a sequence-to-sequence (Seq2Seq) based on Hou et al. (2018) and a VAE based methods from Yoo et al. (2019). Results in Table 2 (*test set*) show that lightweight augmentation is beneficial for both Bi-LSTM CRF and BERT, on both ATIS (single domain) and SNIPS (multi-domain) datasets. SLOT-SUB yields the best results for both the BiLSTM+CRF and BERT models, with SF performance up to 90.43 on ATIS and 90.66 on SNIPS, and IC performance to 95.49 on ATIS and 97.11 on SNIPS. As for the FB dataset, models only gain marginal improvement across lightweight augmentation. We hypothesize that FB is relatively easy to solve, compared with ATIS and SNIPS, as the slot filling performance of BiLSTM without augmentation already achieves a very high F1 score. The improvement using augmentation is more significant for SF rather than for IC.

Out of all lightweight augmentation methods, SLOT-SUB obtains the best performance, particularly on slot filling on ATIS and SNIPS. The overall best performing configuration is a combination of BERT fine-tuning with SLOT-SUB augmentation. Given limited training data, BERT fine-tuning without augmentation surpasses BiLSTM-CRF without augmentation by a large margin. Yet, performance can be boosted even further with lightweight augmentation, suggesting that even a big, self-supervised model, such as BERT can still benefit from augmentation on limited data settings. The improvements on BiLSTM-CRF indicate that lightweight augmentation improves the model's robustness when trained on small amounts of data. We find that SLOT-SUB-LM is suboptimal for SF. Our qualitative observation shows that SLOT-SUB-LM often generates slot values that are semantically incompatible with the original slot label. CROP and ROTATE can help IC in some cases, although their improvement is marginal.

Despite its simplicity, SLOT-SUB is also competitive with state-of-the-art heavyweight data augmentation approaches (Seq2Seq and CVAE), significantly boosting Bi-LSTM and BERT performance for SF on ATIS and SNIPS. We believe that the key advantage of SLOT-SUB is its capability to maintain semantic consistency over the slot spans, which has revealed to be stronger than that of heavyweight approaches. This also shows that slot consistency is crucial for obtaining good performance, particularly for SF. While the CVAE based approach from Yoo et al. (2019) has injected slot and intent labels in the model, it seems that generating semantically consistent utterances is still challenging for deep learning models, especially when data is limited.

4 Analysis and Discussion

In this Section we discuss several aspects of data augmentation applied to slot filling and intent detection.

Impact of number of augmented sentences. To better understand the effect of the number of augmented sentences per utterance (N), we now

Model	DA	ATIS		SNIPS		FB	
		Slot	Intent	Slot	Intent	Slot	Intent
BiLSTM+CRF	None	86.83	90.64	84.51	95.94	93.83	98.47
	Seq2Seq (Hou et al., 2018)	88.72	-	-	-	-	-
	VAE (Yoo et al., 2019)	89.27	90.95	-	-	-	-
	Slot-Sub	89.89†	93.37†	86.45†	96.30†	93.70	98.45
	Slot-Sub-LM	87.03	92.96†	82.82	96.14	91.52	98.20
	Slot-Sub-LM+Filter	87.19	92.01†	82.77	96.08	92.18	98.37
	Crop	88.62†	92.32†	85.84†	96.07	93.91	98.64
	Rotate	88.83†	92.33†	85.65	96.39†	94.04	98.56
BERT	None	89.39	94.98	89.17	96.70	94.22	98.61
	Slot-Sub	**90.43**†	**95.49**†	**90.66**†	**97.11**†	94.01	98.59
	Slot-Sub-LM	87.88	94.49	85.65	96.59	91.84	98.47
	Slot-Sub-LM+Filter	88.37	94.57	86.23	96.60	92.60	98.59
	Crop	89.47	94.55	89.77	96.78	94.20	98.73
	Rotate	89.57	94.48	89.37	96.81	**94.32**	**98.80**

Table 2: Overall results on the test set. Underlined numbers indicate best performing methods for a particular slot filling + intent model. **Bold** numbers indicate best overall methods. † indicates significant improvement over the baseline without augmentation (p-value $<$ 0.05, Wilcoxon signed rank test).

observe the performance of our best performing method, Slot-Sub, while changing N values (we use $\{2, 5, 10, 20, 25\}$) on the dev set. As for ATIS, increasing N yields a F1 improvement from 90.68 up to 91.62; SNIPS performance increased from 87 F1 and to 88 F1 when increasing N from 2 to 5 and it is stable around 88 F1 when using N larger than 5; finally, FB is stable around 93.4 to 93.7 F1. Overall, the biggest improvement is when N is increased from 2 to 5, while with higher values only minor improvements can still be obtained on ATIS.

Performance on different training data size ($\mathcal{D}$). Figure 3 displays the gain obtained by Slot-Sub for various data size for slot filling. Using smaller data size (i.e., 5%) than our default setting, Slot-Sub still obtains a F1 gain for all datasets. On the other hand, as we increase the number of training data, the Slot-Sub benefit diminishes, without hurting performance on ATIS and SNIPS. As for FB we observe a performance drop of less than 1 F1, which is still relatively low.

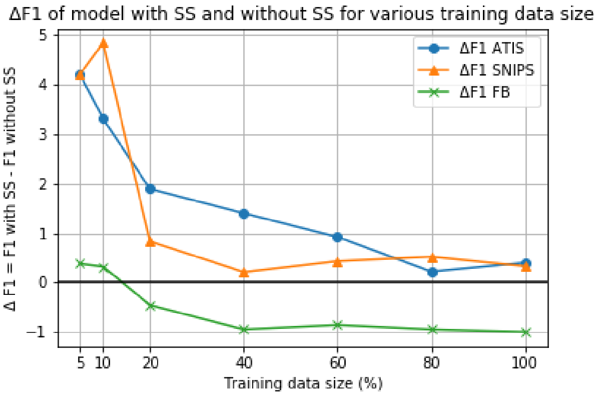


Figure 3: Gain ($\Delta F1$) obtained by Slot-Sub (SS) on various training data size. Positive numbers mean that the model with SS is better than without SS.

Is lightweight augmentation beneficial to very large language models? Motivated by the encouraging results that lightweight augmentation has obtained on a strong pre-trained LM such as BERT on low-resource settings (see Table 2), we now further examine the advantage of lightweight augmentation for other very large pre-trained LM models, namely Albert (Lan et al., 2020) and Roberta (Liu et al., 2019). We use the largest trained models for each of the

Model	Aug.	ATIS		SNIPS	
		Slot	Intent	Slot	Intent
BERT (large)	None	91.6	95.0	89.8	95.0
	SS	92.8	95.4	92.8	95.4
Albert (xxl)	None	92.1	94.8	89.5	99.0
	SS	92.9	95.0	**93.6**	**99.2**
Roberta (large)	None	90.6	92.8	89.2	98.9
	SS	**93.2**	**95.9**	92.5	98.8

Table 3: Lightweight augmentation Slot-Sub (SS) applied to very large pre-trained LMs.

pre-trained LM, namely `bert-large-uncased`, `roberta-large`, and `albert-xxl`. Results, reported in Table 3, show that on limited data settings, all the very large models still benefit from Slot-Sub, notably on the performance for SF.

Qualitative Analysis of slot values from Slot-Sub vs Slot-Sub-LM. The performance of Slot-Sub especially in SF is better than Slot-Sub-LM, as Slot-Sub maintains semantic consistency on the span level. We observe that Slot-Sub-LM often generates slot values that fit the sentence context but that do not maintain the semantics of the slots, which hampers the performance in SF (Table 4). The fact that Slot-Sub-LM often generates "wrong" slot values makes Slot-Sub-LM+Filter also less effective. A possible future direction is to cast Slot-Sub-LM as a *conditional* NLG problem, incorporating labels at the token-level, although this is still challenging when data is limited.

5 Related Work

Data augmentation methods have been widely applied in computer vision, ranging from geometric transformations (Krizhevsky et al., 2012; Zhong et al., 2020), data mixing (Summers and Dinneen, 2019), to the use of generative models (Goodfellow et al., 2014) for generating synthetic data. Recently, data augmentation has been applied to various NLP tasks, including text classification (Wei and Zou, 2019; Wang and Yang, 2015), parsing (Sahin and Steedman, 2018; Vania et al., 2019), and machine translation (Fadaee et al., 2017). Augmentation techniques for NLP tasks range from operations on tokens (e.g., substituting, deleting) (Wang and Yang, 2015; Kobayashi, 2018; Wei and Zou, 2019), to manipulation of the sentence structure (Sahin and Steedman, 2018), to paraphrase-based augmentation (Callison-Burch et al., 2006).

Data augmentation has been also experimented in the context of slot filling and intent classification. Particularly, recent methods have focused on the application of generative models to produce synthetic utterances. Hou et al. (2018) proposes a method that separates the utterance generation from the slot values realization. A sequence to sequence based model is used to generate utterances for a given intent with slot values placeholders (i.e., delexicalized), and then words in the training data that occur in similar contexts of the placeholder are inserted as the slot values. Zhao et al. (2019) also uses a sequence to sequence model by exploiting a small number of template exemplars. Yoo et al. (2019) proposes a solution based on Conditional Variational Auto Encoder (CVAE) to generate synthetic utterances. In this case the CVAE takes into account both the intent and the slot labels during training, and the model generates the surface form of the utterance, slot labels, and the intent label. Recent work from Peng et al. (2020) make use of GPT-2 (Radford et al., 2019), and fine-tuned it to intent and slot-value pairs to generate utterances.

In comparison to existing, state of the art, augmentation methods for slot filling and intent detection, the augmentation methods proposed in this paper can be considered as *lightweight* because they do not require any separate training based on deep learning models for generating additional data. Still, lightweight augmentation maintains consistent slot semantic substitutions, a feature that is crucial for effective data augmentation. In the spectrum of existing augmentation methods, i.e., from words manipulation to paraphrasing-based methods, our lightweight approaches lie in the middle, as we focus either on particular *text spans* that convey slot values, or on particular structures in the dependency parse tree of the utterance.

6 Conclusion

We showed that lightweight augmentation for slot filling and intent detection in low-resource settings is very competitive with respect to more com-

Dataset	Slot	Original Sentence	Slot-Sub	Slot-Sub-LM
ATIS	Depart_Time	List all flights leaving Denver on Continental on Sunday after **934 pm**	List all flights leaving Denver on Continental on Sunday after **7 pm**	List all flights leaving Denver on Continental on Sunday after **Christmas day**
	FromLoc_CityName	List all flights leaving **Denver** on Continental on Sunday after 934 pm	List all flights leaving **Atlanta** on Continental on Sunday after 934 pm	List all flights leaving **Boston** on Continental on Sunday after 934 pm
	Airlines_Name	I need a flight on **Air Canada** from Toronto to San Diego with a layover in DC	I need a flight on **Northwest Airlines** from Toronto to San diego with a layover in DC	I need a flight on **a Thursday** from Toronto to San Diego with a layover in DC
SNIPS	Condition_Description	Will it be **sunny** in Eyota Hawaii on February seventh 2025	Will it be **humid** in Eyota Hawaii on February seventh 2025	Will it be **held** in Eyota Hawaii on February seventh 2025
	Spatial_Relation	What is the **closest** cinema today playing animated movies	What is the **close-by** cinema today playing animated movies	What is the **underground** cinema today playing animated movies
	Restaurant_Type	I need to book a **pub** in Cammack village Wyoming for a party of seven	I need to book a **fast food restaurant** in Cammack village Wyoming for a party of seven	i need to book a **lodge** in Cammack village Wyoming for a party of seven
FB	Date_Time	Set alarm **for 4 am** tomorrow morning	Set alarm **at 6** tomorrow morning	Set alarm **for me** tomorrow morning
	Location	How hot is it in **Hong Kong** ?	How hot is it in **Fairbanks** ?	How hot is it in **the mornings** ?

Table 4: Samples of sentences from Slot-Sub and Slot-Sub-LM. The bold text span denotes the span that is substituted. The text span in blue denotes semantically consistent replacements, while red indicates semantically inconsistent substitutes.

plex deep learning based data augmentation. A lightweight method based on slot values substitution, while preserving the semantic consistency of slot labels, has proven to be the more effective. We also show that large self-supervised models, like BERT, can benefit from lightweight augmentation, suggesting that a *combination* of data augmentation and transfer learning is very useful, and has the potential to be applied to other NLP tasks.

As for future work, it would be interesting to see the effect of using the augmented data generated by Slot-Sub as additional training data for deep learning based augmentation models. Encouraged by the results of our lightweight augmentation, our work can also be experimented on semantic tasks with similar characteristics, such as Named Entity Recognition.

References

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 17–24, New York City, USA, June. 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. *ArXiv*, abs/1805.10190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2492–2501. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada, July. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1195–1205. Association for Computational Linguistics.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The atis spoken language systems pilot corpus. In *HLT*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1234–1245. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 452–457. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentence-level information with encoder LSTM for semantic slot filling. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2077–2083. The Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Z. Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23:530–539.

Baolin Peng, Chenguang Zhu, Michael Zeng, and Jianfeng Gao. 2020. Data augmentation for spoken language understanding via pretrained models. *CoRR*, abs/2004.13952.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2078–2087. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Gözde Gül Sahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5004–5009. Association for Computational Linguistics.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2018. Cross-lingual transfer learning for multilingual task oriented dialog. In *NAACL-HLT*.

Cecilia Summers and Michael J Dinneen. 2019. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1262–1270. IEEE.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Clara Vania, Yova Kementchedjhieva, Anders Søgaard, and Adam Lopez. 2019. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1105–1116. Association for Computational Linguistics.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2557–2563. The Association for Computational Linguistics.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6381–6387. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2019. Data augmentation for spoken language understanding via joint variational generation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7402–7409. AAAI Press.

Zijian Zhao, Su Zhu, and Kai Yu. 2019. Data augmentation with atomic templates for spoken language understanding. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3635–3641. Association for Computational Linguistics.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2020. Random erasing data augmentation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13001–13008. AAAI Press.

Appendix A. Hyperparameters

Hyperparameter	Value
Learning rate	10^{-5}
Dropout	0.1
Mini-batch size	16
Optimizer	BertAdam
Number of epoch	30 (`bert-base-uncased`) 10 (`bert-large`, `roberta-large`, `albert-xxl`)
Early stopping	10
nb_{aug}	Tuned on $\{2, 5, 10\}$
Nucleus sampling	top-*p* = 0.9
Max rotation	3
Max crop	3

Table 5: Hyperparameters used for the Transformer based models and data augmentation methods

Appendix B. Training Data for SLOT-SUB-LM+Filter

Dataset	#train
ATIS	7,846
SNIPS	24,472
FB	52,798

Table 6: Total training examples for SLOT-SUB-LM+Filter. The number of positive and negative examples are the same.