Learning from Mistakes: Combining Ontologies via Self-Training for Dialogue Generation

Lena Reed[1], **Vrindavan Harrison**[1], **Shereen Oraby**[2*], **Dilek Hakkani-Tür**[2*], **and Marilyn Walker**[1]
[1]Natural Language and Dialogue Systems Lab, University of California, Santa Cruz
[2]Amazon Alexa AI
{lireed,vharriso,mawalker}@ucsc.edu
{orabys,hakkanit}@amazon.com

Abstract

Natural language generators (NLGs) for task-oriented dialogue typically take a meaning representation (MR) as input, and are trained end-to-end with a corpus of MR/utterance pairs, where the MRs cover a specific set of dialogue acts and domain attributes. Creation of such datasets is labor intensive and time consuming. Therefore, dialogue systems for new domain ontologies would benefit from using data for pre-existing ontologies. Here we explore, for the first time, whether it is possible to train an NLG for a new **larger** ontology using existing training sets for the restaurant domain, where each set is based on a **different** ontology. We create a new, larger **combined** ontology, and then train an NLG to produce utterances covering it. For example, if one dataset has attributes for *family friendly* and *rating* information, and the other has attributes for *decor* and *service*, our aim is an NLG for the combined ontology that can produce utterances that realize values for *family friendly*, *rating*, *decor* and *service*. Initial experiments with a baseline neural sequence-to-sequence model show that this task is surprisingly challenging. We then develop a novel **self-training** method that identifies (errorful) model outputs, automatically constructs a corrected MR input to form a new (MR, utterance) training pair, and then repeatedly adds these new instances back into the training data. We then test the resulting model on a new test set. The result is a self-trained model whose performance is an absolute 75.4% improvement over the baseline model. We also report a human qualitative evaluation of the final model showing that it achieves high naturalness, semantic coherence and grammaticality.

1 Introduction

Natural language generators (NLGs) for task-oriented dialogue take meaning representations (MRs) as inputs, i.e. a set of dialogue acts with attributes and their values, and output natural language utterances realizing the MR. Current NLGs are trained end-to-end with a corpus of MR/utterance pairs where the MRs cover a specific set of dialogue acts and domain attributes. Creation of such datasets is labor intensive and time consuming. However, when building an NLG for a new domain ontology, it should be possible to re-use data built on existing domain ontologies. If this were possible, it would speed up development of new dialogue systems significantly.

Here we experiment with one version of this task by building a new domain ontology based on **combining** two existing ontologies, and utilizing their training data. Each dataset is based on a different domain ontology in the restaurant domain, with novel attributes and dialogue acts not seen in the other dataset, e.g. only one has attributes representing *family friendly* and *rating* information, and only one has attributes for *decor* and *service*. Our aim is an NLG engine that can realize utterances for the extended **combined** ontology not seen in the training data, e.g. for MRs that specify values for *family friendly*, *rating*, *decor* and *service*. Figure 1 illustrates this task. Example E1 is from a training set referred to as NYC, from previous work on controllable sentence planning in NLG (Reed et al., 2018), while E2 is from the E2E NLG shared task (Novikova et al., 2017a). As we describe in detail in Section 2, E1 and E2 are based on two distinct ontologies. Example E3 illustrates the task addressed in this paper: we create a test set of novel MRs for the combined ontology, and train a model to generate high quality outputs where individual sentences realize attributes from both ontologies.

To our knowledge, this is a completely novel task. While it is common practice in NLG to construct test sets of MRs that realize attribute combinations not seen in training, initial experiments

*Work done prior to joining Amazon.

Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 314 - 327
July 1-3, 2020

ID	Ontology	MEANING REPRESENTATION	EXAMPLE
E1	NYC (TRAINING)	RECOMMEND[YES], INFORM(NAME[RESTAURANT], SERVICE[EXCELLENT], FOOD[EXCELLENT], DÉCOR[EXCELLENT], LOCATION[AREA], PRICE[EXPENSIVE])	I suggest you go to [RESTAURANT]. The food, service and atmosphere are all excellent, even if it is expensive. Its in [AREA].
E2	E2E (TRAINING)	INFORM(NAME[RESTAURANT], EATTYPE[RESTAURANT-TYPE], CUSTOMER-RATING[HIGH], AREA[AREA], NEAR[POINT-OF-INTEREST])	[RESTAURANT] is a [RESTAURANT-TYPE] in [AREA] near [POINT-OF-INTEREST]. It has a high customer rating.
E3	COMBINED (TEST)	RECOMMEND = YES, INFORM(NAME[RESTAURANT], EATTYPE[RESTAURANT-TYPE], FOOD = EXCELLENT, LOCATION[AREA], NEAR[POINT-OF-INTEREST], CUSTOMER-RATING[HIGH], DÉCOR = EXCELLENT, SERVICE=EXCELLENT, PRICE=EXPENSIVE)	[RESTAURANT] is the best because it has excellent service and atmosphere. It is a [RESTAURANT-TYPE] offering excellent food in [AREA] near [POINT-OF-INTEREST] with a high customer rating, but it is expensive.

Figure 1: E1 and E2 illustrate training instances from the two source datasets E2E and NYC. E2E attributes are represented in blue and NYC is in red. Some attributes are shared between both sources: here the unique dialogue acts and attributes for each source are underlined in E1 and E2. E3 illustrates an MR from the target test set that we dub COM. All the MRs in COM combine dialogue acts and attributes from E2E and NYC. There is no training data corresponding to E3. The MRs illustrate how some attribute values, e.g. RESTAURANT NAME, POINT-OF-INTEREST, are delexicalized to improve generalization.

showed that this task is surprisingly adversarial. However, methods for supporting this type of generalization and extension to new cases would be of great benefit to task-oriented dialogue systems, where it is common to start with a restricted set of attributes and then enlarge the domain ontology over time. New attributes are constantly being added to databases of restaurants, hotels and other entities to support better recommendations and better search. Our experiments test whether existing data that only covers a subset of attributes can be used to produce an NLG for the enlarged ontology.

We describe below how we create a test set — that we call COM — of combined MRs to test different methods for creating such an NLG. A baseline sequence-to-sequence NLG model has a slot error rate (SER) of .45 and only produces semantically perfect outputs 3.5% of the time. To improve performance, we experiment with three different ways of conditioning the model by incorporating *side constraints* that encode the source of the attributes in the MR (Sennrich et al., 2016; Harrison et al., 2019). However, this only increases the proportion of semantically perfect model outputs from 3.5% to 5.5% (Section 4.1).

We then propose and motivate a novel self-training method that greatly improves performance by learning from the model mistakes. An error analysis shows that the models **do** produce many **combined** outputs, but with errorful semantics. We develop a rule-based text-to-meaning semantic extractor that automatically creates novel correct MR/text training instances from errorful model outputs, and use these in self-training experiments, thus learning from our mistakes (Section 4.2). We validate the text-to-meaning extractor with a human evaluation. We find that a model trained with this process produces SERs of only .03, and semantically perfect outputs 81% of the time (a 75.4 percent improvement). A human evaluation shows that these outputs are also natural, coherent and grammatical. Our contributions are:

- Definition of a novel generalization task for neural NLG engines, that of generating from unseen MRs that combine attributes from two datasets with different ontologies;
- Systematic experiments on methods for conditioning NLG models, with results showing the effects on model performance for both semantic errors and combining attributes;
- A novel self-training method that learns from the model's mistakes to produce semantically correct outputs 81% of the time, an absolute 75.4% improvement.

We start in Section 2 by defining the task in more detail, describe our models and metrics in Section 3, and results in Section 4. We discuss related work throughout the paper where it is most relevant and in the conclusion in Section 5.

2 Ontology Merging and Data Curation

We start with two existing datasets, NYC and E2E, representing different ontologies for the restaurant

domain. The NYC dataset consists of 38K utterances (Reed et al., 2018; Oraby et al., 2018), based on a restaurant ontology used by Zagat (Stent et al., 2002, 2004).[1] The E2E dataset consists of 47K utterances distributed for the E2E Generation Challenge (Novikova et al., 2017a).[2] Each dataset consists of pairs of reference utterances and meaning representations (MRs). Figure 1 shows sample MRs for each source and corresponding training instances as E1 and E2.

Ontology Merging. We first make a new combined ontology ONTO-COM by merging NYC and E2E. Attributes, dialogue acts, and sample values for E2E and NYC are illustrated on the left-hand side of Figure 2, and the result of merging them to create the new ontology is on the right-hand side of Figure 2. Since there are only 8 attributes in each source dataset, we developed a script by hand that maps the MRs from each source into the ONTO-COM ontology.

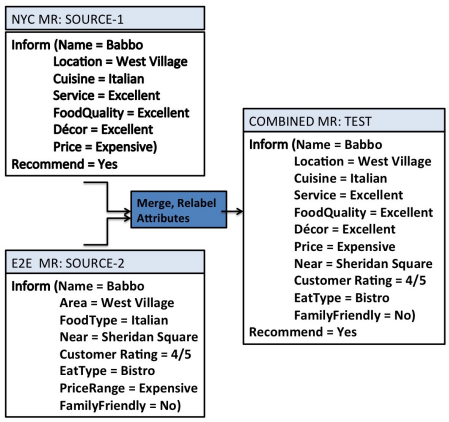

Figure 2: An example illustrating how dialogue acts and attributes for both source databases are merged and relabelled to make a new combined ontology used in train and test.

As Figure 2 shows, both datasets have the INFORM dialogue act, and include the attributes *name*, *cuisine*, *location*, and *price* after mapping. The unique attributes for the NYC ontology are scalar ratings for *service*, *food quality* and *decor*. The NYC dataset also has the RECOMMEND dialogue act, seen in E1 in Figure 1. The unique attributes of the E2E ontology are *customer rating*, *eat type* ("coffee shop"), *near* and *family friendly*.

Training Data. Given the combined ontology ONTO-COM, we then map the training data for both E2E and NYC into ONTO-COM by relabelling the MRs to have consistent names for shared attributes as illustrated in Figure 2. We create a balanced training set of ∼77K from the two original datasets by combining all NYC references with a random same-size sample of E2E references.

Test Set. We then manually create a test set, COM, consisting of 3040 MRs based on the new combined ontology ONTO-COM. Each test MR must have at least one attribute from E2E and one attribute from NYC so that it combines attributes from both sources: these MRs provide combinations never seen in training.[3] Example E3 in Figure 1 provides an example test MR. The procedure for creating the test set ensures that the length and complexity of the test set are systematically varied, with lengths normally distributed and ranging from 3 to 10 attributes. Recommendations only occur in the NYC training data, and they increase both **semantic** and **syntactic** complexity, with longer utterances that use the discourse relation of JUSTIFICATION (Stent et al., 2002), e.g. *Babbo is the best* **because** *it has excellent food*. We hypothesize that recommendations may be more challenging to combine across domains, so we vary MR complexity by including the RECOMMEND dialogue act in half the test references. We show in Section 4 that the length and complexity of the MRs is an important factor in the performance of the trained models.

3 Experimental Overview and Methods

Given the training and test sets for the combined ontology in Section 2, we test 4 different neural model architectures and present results in Section 4.1. We then propose a a novel self-training method, and present results in Section 4.2. These experiments rely on the model architectures presented here in Section 3.1, and the Text-to-Meaning semantic extractor and performance metrics in Section 3.2.

3.1 Model Architectures

In the recent E2E NLG Challenge shared task, models were tasked with generating surface forms from structured meaning representations (MRs) (Dušek et al., 2020). The top performing models were all RNN encoder-decoder systems. Here we also use a standard RNN Encoder–Decoder model (Sutskever et al., 2014) that maps a source sequence (the input MR) to a target sequence (the utterance text). We

[1] http://nlds.soe.ucsc.edu/sentence-planning-NLG
[2] http://www.macs.hw.ac.uk/InteractionLab/E2E/
[3] The train and test data are available at http://nlds.soe.ucsc.edu/source-blending-NLG

first implement a baseline model and then add three variations of model supervision that aim to improve semantic accuracy. All of the models are built with OpenNMT-py, a sequence-to-sequence modeling framework (Klein et al., 2017).

Encoder. The MR is represented as a sequence of (attribute, value) pairs with separate vocabularies for attributes and values. Each attribute and each value are represented using 1-hot vectors. An (attribute, value) pair is represented by concatenating the two 1-hot vectors.

The input sequence is processed using two single layer bidirectional-LSTM (Hochreiter and Schmidhuber, 1997) encoders. The first encoder operates at the pair level, producing a hidden state for each attribute-value pair of the input sequence. The second LSTM encoder is intended to produce utterance level context information in the form of a full MR encoding produced by taking the final hidden state after processing the full input sequence. The outputs of both encoders are combined via concatenation. That is, the final state of the second encoder is concatenated onto each hidden state output by the first encoder. The size of the pair level encoder is 46 units and the size of the MR encoder is 20 units. Model parameters are initialized using Glorot initialization (Glorot and Bengio, 2010) and optimized using Stochastic Gradient Descent with mini-batches of size 128.

Decoder. The decoder is a uni-directional LSTM that uses global attention with input-feeding. Attention weights are calculated via the *general* scoring method (Luong et al., 2015). The decoder takes two inputs at each time step: the word embedding of the previous time step, and the attention weighted average of the encoder hidden states. The ground-truth previous word is used when training, and the predicted previous word when evaluating. Beam search with five beams is used during inference.

Supervision. Figure 3 shows the baseline system architecture as well as three types of supervision, based on conditioning on source (E2E, NYC) information. The additional supervision is intended to help the model attend to the source domain information. We call the three types of supervision GUIDE, ATTR and BOOL, and the baseline architecture NOSUP, representing that it has no additional supervision.

The supervision methods are shown in Figure 4. The source feature has a vocabulary of three items: *nyc*, *e2e* and *both*. Since *both* is never seen

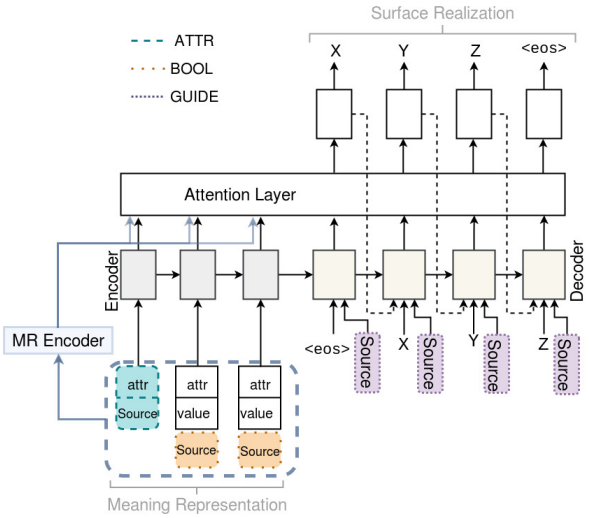

Figure 3: Attentional Encoder-Decoder architecture with each supervision method shown.

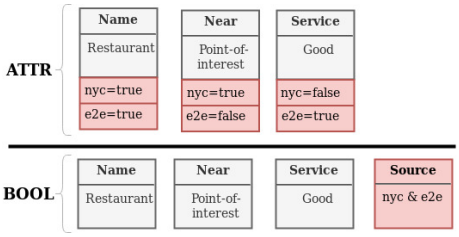

Figure 4: An illustration of ATTR and BOOL supervision methods, with the source supervision (NYC or E2E) shown in red.

in train, the source information is represented using two booleans: *True*||*False* denotes a reference from E2E while *False*||*True* denotes a reference from NYC. This encoding is intended to encourage generalization at inference time. During inference, blending of information from both sources is specified by using *True*||*True*. The ATTR supervision method represents the source information by concatenating the boolean source token onto each attribute as seen in Figure 4. This redundantly represents the source information locally to each attribute, which has been effective for tasks such as question generation and stylistic control (Harrison and Walker, 2018; Harrison et al., 2019). The BOOL supervision method adds the boolean source token to the end of the sequence of attribute-value pairs as its own attribute, as in work on machine translation and controllable stylistic generation (Sennrich et al., 2016; Yamagishi et al., 2016; Ficler and Goldberg, 2017). The GUIDE model inputs the source information directly to the decoder LSTM. In previous work, putting information into the decoder in this way has yielded improvements in paraphrase

generation and controllable generation (Iyyer et al., 2018; Harrison et al., 2019)

3.2 Text-to-Meaning Semantic Extractor

Much previous work in NLG relies on a test set that provides gold reference outputs, and then applies automatic metrics such as BLEU that compare the gold reference to the model output (Papineni et al., 2002; Dušek et al., 2020), even though the limitations of BLEU for NLG are widely acknowledged (Belz and Reiter, 2006; Stent et al., 2005; Novikova et al., 2017b; Liu et al., 2016). To address these limitations, recent work has started to develop "referenceless" NLG evaluation metrics (Dusek et al., 2017; Kann et al., 2018; Tian et al., 2018; Mehri and Eskenazi, 2020).

Since there are no reference outputs for the COM test set, we need a referenceless evaluation metric. We develop a rule-based text-to-MR semantic extractor (TTM) that allows us to compare the input MR to an MR automatically constructed from an NLG model textual output by the TTM, in order to calculate **SER**, the slot error rate. The TTM system is based on information extraction methods. We conduct a human evaluation of its accuracy below. A similar approach is used to calculate semantic accuracy in other work in NLG, including comparative system evaluation in the E2E Generation Challenge (Juraska et al., 2018; Dušek et al., 2020; Wiseman et al., 2017; Shen et al., 2019).

The TTM relies on a rule-based automatic aligner that tags each output utterance with the attributes and values that it realizes. The aligner takes advantage of the fact that the RECOMMEND dialogue act, and the attributes and their values are typically realized from a domain-specific finite vocabulary. The output of the aligner is then used by the TTM extractor to construct an MR that matches the (potentially errorful) utterance that was generated by the NLG. We refer to this MR as the "retrofit MR". The retrofit MR is then compared to the input MR in order to automatically calculate the slot error rate **SER**:

$$SER = \frac{D + R + S + H}{N}$$

where D is the number of deletions, R is the number of repetitions, S is the number of substitutions, H is the number of hallucinations and N is the number of slots in the input MR (Nayak et al., 2017; Reed et al., 2018; Wen et al., 2015). Section A.1 in the supplementary materials provides more detail and examples for each type of semantic error. SER is first calculated on individual utterances and then averaged over the whole test set. For additional insight, we also report the percentage of **semantically perfect outputs** (perfect%), outputs where the SER is 0 and there are no semantic errors. This measure is analogous to the Sentence Error Rate used in speech recognition.

Human TTM Accuracy Evaluation. We evaluated the TTM and the automatic SER calculation with a separate experiment where two NLG experts hand-labelled a random sample of 200 model outputs. Over the 200 samples, the automatic SER was .45 and the human was .46. The overall correlation of the automatic SER with the human SER over all types of errors (D,R,S,H) is .80 and the correlation with deletions, the most frequent error type, is .97.

Retrofit MRs for Self-Training. The TTM is critical for our novel self-training method described in Section 4.2. The retrofit MRs match the (errorful) NLG output: when these MR/NLG output pairs combine attributes from both sources, they provide novel corrected examples to add back into training.

4 Results

We run two sets of experiments. We first run all of the NLG models described in Section 3.1 on the COM test set, and automatically calculate SER and perfect% as described in Section 3.2. We report these results in Section 4.1. Section 4.2 motivates and describes the self-training method and presents the results, resulting in final models that generate semantically perfect outputs 83% of the time.

4.1 Initial Model Results

Model	Training	Test	SER	PERFECT	
				N	%
NOSUP	E2E + NYC	COM	**.45**	106	3.5%
GUIDE	E2E + NYC	COM	.66	15	0.5%
ATTR	E2E + NYC	COM	.46	167	**5.5%**
BOOL	E2E + NYC	COM	**.45**	86	2.8%

Table 1: SER and perfect% on test for each model type on the test of 3040 MRs (COM) that combine attributes from both sources.

Semantic Accuracy. Table 1 summarizes the results across the four models NOSUP, GUIDE, ATTR and BOOL. Overall, the results show that the task, and the COM test set, are surprisingly adversarial. All of the models have extremely high SER, and the SER for NOSUP, ATTR, and BOOL are very similar. Row 2 shows that the GUIDE model has much worse performance than the other models,

in contrast to other tasks (Iyyer et al., 2018). We do not examine the GUIDE model further. Row 3 shows that the ATTR supervision results in the largest percentage of perfect outputs (5.5%).

Model	Training	Test	SER	PERF %
NOSUP	E2E	E2E	.16	19%
NOSUP	E2E + NYC	E2E	.18	15%
NOSUP	NYC	NYC	.06	69%
NOSUP	E2E + NYC	NYC	.06	71%

Table 2: Baseline results for each source on its own test using the NOSUP model. E2E test N = 630. NYC test N = 314.

The results in Table 1 should be compared with the baselines for testing NOSUP on **only** E2E or NYC in Table 2. Both the E2E and NYC test sets consist of unseen inputs, where E2E is the standard E2E generation challenge test (Dušek et al., 2020), and NYC consists of novel MRs with baseline attribute frequencies matching the training data.[4] Rows 1 and 3 test models trained on only E2E or only NYC, while Rows 2 and 4 test the same trained NOSUP model used in Row 1 of Table 1 on E2E or NYC test sets respectively. Comparing Rows 1 and 2 shows that training on the same combined data used in Table 1 slightly degrades performance on E2E, however, this SER is still considerably lower than the .45 SER for the NOSUP model tested on the COM test set, shown in the first row of Table 1. Row 4 shows that the NOSUP model trained on the combined data appears to improve performance on the NYC test because the perfect% goes up from 69% in Row 3 to 71%. The SER of .06 shown in Row 4 should also be compared to the .45 SER reported for the NOSUP model in the first row of Table 1. These results taken together establish that the combined MRs in the COM test provide a very different challenge than the E2E and NYC unseen test inputs.

However, despite the poor performance of the initial models, we hypothesized that there may be enough good outputs to experiment with self-training. Since the original training data had no combined outputs, decoding may benefit from even small numbers of training items added back in self-training.

Human Evaluation. The automatic SER results provide insight into the semantic accuracy of the models, but no assessment of other aspects of performance. We thus conduct a human evaluation on Mechanical Turk to qualitatively assess fluency, coherency and grammaticality. We use the automatic SER to select 100 semantically perfect references from the NOSUP and the ATTR models' test outputs, and the 86 perfect references from BOOL. We ask 5 Turkers to judge on a scale of 1 (worst) to 5 (best) whether the utterance is: (1) fluent and natural; (2) semantically coherent; and (3) grammatically well-formed. Table 3 reports the average score for these qualitative metrics as well as the Turker agreement, using the average Pearson correlation across the Turkers. The results show that the agreement among Turkers is high, and that all the models perform well, but that the ATTR model outputs are the most natural and coherent, while the BOOL model outputs are the most grammatical.

Model	NAT.	COHER.	GRAMMAT.
NOSUP	4.04	4.13	4.12
ATTR	**4.11**	**4.25**	4.14
BOOL	3.97	4.18	**4.25**
AGREEMENT	.63	.62	.65

Table 3: Human Evaluation for NOSUP (N = 100) ATTR (N = 100) and BOOL (N = 86) for Naturalness, Semantic Coherence, and Grammaticality

4.2 Self-Training

In order to conduct self-training experiments, we need perfect outputs that combine attributes from both sources to add back into training. These outputs must also be natural, coherent and grammatical, but Table 3 shows that this is true of all the models. A key idea for our novel self-training method is that the TTM (Section 3.2) automatically produces "retrofit" corrected MRs that match the output texts of the NLG models. Thus we expect that we can construct more perfect outputs for self-training by using retrofitting than those in Table 1. Here, we first analyse the outputs of the initial models to show that self-training is feasible, and then explain our method and present results.

Error Analysis. An initial examination of the outputs suggests that the models simply have trouble combining attributes from both sources. We provide examples in Table 10 in Section A.2 in the supplementary materials. To quantify this observation, we define a metric, Source Blending Rate (**SB**), that counts the percentage of outputs that combine attributes from both sources, whether or

[4] Previous work on the E2E dataset has also used seq2seq models, with SOA results for SER of 1% (Dušek et al., 2020), but here we do not use the full training set. Our partition of the NYC dataset has not been used before, but experiments on comparable NYC datasets have SERs of .06 and .02 (Reed et al., 2018; Harrison et al., 2019).

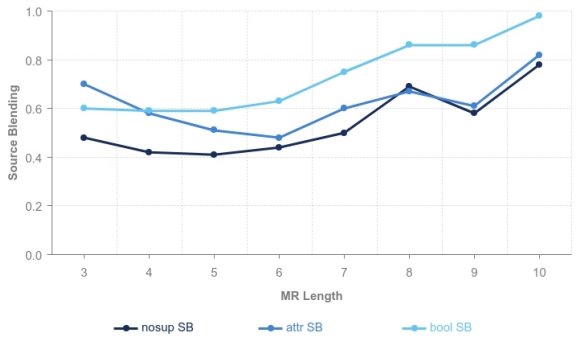

Figure 5: Source Blending Rate (SB) as a function of MR length for NOSUP, ATTR and BOOL.

not the attribute values are accurate:

$$SB = \frac{R_{sb}}{N}$$

where R_{sb} is the count of references r that contain an attribute $a_i \subseteq source_1$ and another attribute $a_j \subseteq source_2$, and N is the total number of references. Only attributes that appear uniquely in each source are included in the a_i, a_j: the unique attributes are illustrated in Figure 2.

Figure 5 graphs SB as a function of MR length showing that indeed the models **do** in many cases produce combined outputs and that the type of model supervision greatly influences SB. The NOSUP model is the worst: a fact that is masked by the NOSUP model's SER in Table 1, which appears to be on a par with both ATTR and BOOL. Interestingly, all models are more likely to produce an SB output as the MRs get longer, but Figure 5 shows clearly that the BOOL model especially excels.

For self-training, we also need a model that generates utterances with the RECOMMEND dialogue act. As mentioned in Section 2, recommendations increase both semantic and syntactic complexity. Half the test items contain a recommendation, so we need a model that can produce them. Table 4 presents results for SER and SB depending on whether a RECOMMEND was in the MR, showing that the three models vary a great deal. However, the BOOL row for the SB column shows that when the MR includes a recommendation, the BOOL model produces a combined output far more frequently than NOSUP or ATTR (SB = .73).

Thus Figure 5 and Table 4 show that the BOOL model produces the most combined outputs. After TTM extraction, the BOOL model provides the most instances (1405) of retrofit MR/output pairs to add to self-training, and we therefore use BOOL in the self-training experiments below.

Retrofitting MRs for Self-Training. Table 5 illustrates how the TTM works, and shows that it can effectively create a new MR that may not have been previously seen in training, allowing the model to **learn from its mistakes**. The caption for Table 5 explains in detail the retrofitting process and how it leads to new examples to use in self-training.

Model	SER		SB	
	REC	NO-REC	REC	NO-REC
NOSUP	.43	.46	.44	.56
ATTR	.51	.41	.36	.77
BOOL	.47	.43	.73	.67

Table 4: Effect of the RECOMMEND dialogue act on Slot Error Rate (SER) and Source Blending (SB) for the three types of model supervision: NOSUP, ATTR and BOOL.

It is important to note that the retrofit MRs for some NLG outputs **cannot** be used for self-training. NLG model outputs whose semantic errors include repetitions can **never** be used in self-training, because valid MRs do not include repeated attributes and values, and the method doesn't edit the NLG output string. However, deletion errors cause no issues: the retrofit MR simply doesn't have that attribute. Substitutions and hallucinations can be used because the retrofit MR substitutes a value or adds a value to the MR, as long as the realized attribute value is valid, e.g. "friendly food" is not a valid value for *food quality*.[5,6]

Experiments. To begin the self-training experiments, we apply the source-blending metric (SB) defined above to identify candidates that combine attributes from both sources, and then apply the TTM to construct MRs that match the NLG model outputs, as illustrated in Table 5, eliminating references that contain a repetition. We start with the same combined 76,832 training examples and the 1405 retrofit MR/NLG outputs from the BOOL model. We explore two bootstrapping regimes, depending on whether a model output is a repetition of one that we have already seen in training. One model keeps repetitions and adds them back into training, which we dub S-Repeat, and the other model only adds unique outputs back into training, which we dub S-Unique.

Quantitative Results. Figure 6 shows how the SER and perfect% continuously improve on the

[5]We applied the human evaluation in Section 3.2 to instances included in self-training: the correlation between human judgements and the automatic SER is .95, indicating that the retrofit MRs are highly accurate.

[6]Table 10 in Section A.2 provides additional examples of errorful outputs that **can** or **cannot** be used in self-training.

Original MR	Text-to-MR	OUTPUT
name[RESTAURANT], cuisine[fastfood], decor[good], qual[fantastic], location[riverside], price[cheap], eatType[pub], familyFriendly[no]	name[RESTAURANT], cuisine[fastfood], qual[good], location[riverside], familyFriendly[no]	[RESTAURANT] is a fast food restaurant located in the riverside area. it has good food and it is not family friendly.
name[RESTAURANT], recommend[yes], cuisine[fastfood], qual[good], location[riverside], familyFriendly[no]	name[RESTAURANT], cuisine[fastfood], qual[good], location[riverside], familyFriendly[no]	[RESTAURANT] is a fast food restaurant in the riverside area. it is not family friendly and has good food.

Table 5: Examples to show retrofitting. The examples start from different original MRs (col 1), but yield the same MR after text-to-MR extraction (col 2). In Row 1, the model output in column 3 deleted the attributes *price*, *decor* and *eat type* (pub), and substituted the value "good" for "fantastic" for the quality attribute. In Row 2 the model deleted the RECOMMEND dialogue act, but otherwise realized the original MR correctly. At test time, the original MRs produced different outputs (col 3). Thus the retrofitting yields two unique novel instances for self-training.

COM test set for S-Repeat over 10 rounds of self-training, and that S-Repeat has better performance, indicating that adding multiple instances of the same item to training is useful. The performance on the COM test set of the S-Unique model flattens after 8 rounds. After 10 rounds, the S-Repeat model has an SER of .03 and produces perfect outputs 82.9% of the time, a 77.4 percent absolute improvement over the best results in Table 1.

Figure 6: SER and perfect% on the COM test set for S-Repeat vs. S-Unique during self-training

COM-2 Test Set. Since the the self-training procedure used the COM test set during self-training, we construct a new test with 3040 novel MRs using the procedure described in Section 2, which we call COM-2. First we test the initial models on COM-2, resulting in a best SER of 0.45 for the BOOL model, identical with the result for COM. For perfect% the best result was 5.3% on the ATTR model, which is again comparable to the original COM test set. We then tested the final self-trained model on COM-2, with the result that the SER for S-Repeat (0.03) and S-Unique (0.11) are again identical to the result for COM. The perfect% is comparable to that reported in Figure 6; it decreases by 2.2% for S-Repeat to 80.7% and increases by .2% for S-Unique to 50.7%. Overall, the performance on COM-2 improved by an absolute 75.4%.

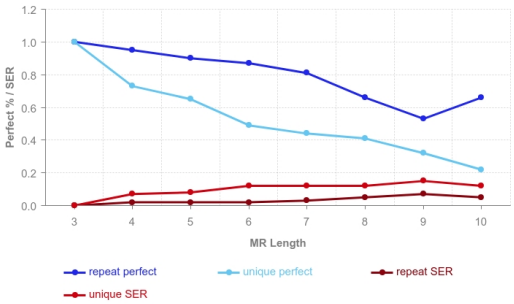

Figure 7: SER and perfect% on COM-2 as a function of MR length for BOOL supervision before self-training and for the S-Repeat model after self-training.

Figure 7 shows that the results improve, not only overall, but also by MR length. It plots the SER and perfect% results, by MR length, for the BOOL model before and after self-training. While the perfect% decreases as the number of attributes increase, there is a large improvement over the initial model results. Also, after self-training the worst perfect% is still above 0.5, which is higher than perfect% for any MR length before self-training. The SER also improves over all MR lengths after self-training, not exceeding .06, significantly better than even the shortest MR before self-training.[7]

Human Evaluation. We also performed a human

Model	NAT.	COHER.	GRAMMAT.
S-REPEAT	3.99	4.08	4.02
S-UNIQUE	**4.06**	**4.13**	**4.14**
AGREEMENT	.57	.61	.57

Table 6: Human Evaluation on Mechanical Turk for S-Repeat (N = 100) and S-Unique (N = 100) for Naturalness, Semantic Coherence, and Grammaticality

[7]Performance results for the self-trained model on the original E2E and NYC test sets in supplement A.3 shows that performance also improves on the E2E and NYC test sets.

evaluation on Mechanical Turk to assess the qualitative properties of the model outputs after self-training. We selected 100 perfect references for S-Repeat and 100 for S-Unique and used the same HIT as described in Section 4.1. Table 6 reports the average score for these qualitative metrics as well as the Turker agreement, using the average Pearson correlation across the Turkers. The results show that naturalness, coherence and grammaticality are still high after self-training for both models, but that the S-Unique model produce better outputs from a qualitative perspective. We believe we could improve the self-training method used here with additional referenceless evaluation metrics that aim to measure naturalness and grammaticality (Mehri and Eskenazi, 2020). We leave this to future work.

#	Realization
1	[RESTAURANT] is the best place because it is a family friendly pub with good decor and good food.
2	[RESTAURANT] is a family friendly restaurant with bland food and is in the low price range. It is the best restaurant.
3	[RESTAURANT] is a family friendly coffee shop with decent service and a low customer rating. It is in the £20-25 price range.
4	[RESTAURANT] is the best restaurant because it is in the east village, it is near [POINT-OF-INTEREST] with great service and it is affordable.

Table 7: Example outputs with source blending. NYC attributes are represented using red and E2E attributes are represented using blue

Qualitative and Linguistic Analysis. Table 7 provides outputs from the models that display different ways of combining attributes from the original sources. In Row 1 we can see that the RECOMMEND dialogue act from NYC can be combined in the same sentence as the attributes *family friendly* and *eat type* from E2E and aggregate these E2E attributes with NYC attributes *decor* and *food quality* using a "with" operator. Row 2 shows another example where the NYC and E2E attributes are joined using a "with" operator. In Row 3 there is a single sentence with four attributes where the NYC attribute is preceded and followed by E2E attributes. Row 4 concatenates the two sources in a single sentence using sentence coordination. The "east village" location from the NYC dataset, is concatenated with the attributes *near* from E2E and *service* from NYC. These examples show that the NLG models can combine attributes from both sources in many different ways. Table 11 in Section A.4 provides additional detail by providing examples along with their corresponding MRs.

5 Conclusion

This paper presents the first experiments on training an NLG for an extended domain ontology by re-using existing within-domain training data. We show that we can combine two training datasets for the restaurant domain, that have different ontologies, and generate output that combines attributes from both sources, by applying a combination of neural supervision and a novel self-training method. While it is common practice to construct test sets with unseen attribute combinations, we know of no prior work based on constructing a new combined ontology. Our experiments show that the task is surprisingly adversarial, consistent with recent work suggesting that neural models often fail to generalize (Wallace et al., 2019; Feng et al., 2018; Ribeiro et al.; Goodfellow et al., 2014). Work on domain transfer shares similar goals to the experiments presented here (Wen et al., 2016; Golovanov et al., 2019), but these methods do not produce NLG outputs that integrate attributes from two different sources into the same sentence. Our final results show that the ability of our self-training method to automatically construct new training instances results in high quality natural, coherent and grammatical outputs with high semantic accuracy.

In future, we hope to generalize our novel self-training method to build an NLG that can combine two distinct domains, e.g. hotels or movies combined with restaurants in multi-domain dialogue (Budzianowski et al., 2018; Gašić et al., 2015; Hakkani-Tür et al., 2016; Cervone et al., 2019; Ultes et al., 2017). Ideally systems that cover multiple domains should be able to produce utterances that seamlessly integrate both domains, if data exists for each domain independently. However, there may be additional challenges in such combinations. Our results require the initial neural models to generate **some** combined outputs. It is not clear whether there are some aspects of our experimental setup that facilitate this, e.g. it may require some attributes to be shared across the two initial ontologies, or some shared vocabulary. Thus it is possible that initial models for two more distinct domains may not produce any combined outputs, and it may be necessary to seed the self-training experiments with a small number of combined training instances. We leave these issues to future work.

References

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of nlg systems. In *EACL*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Alessandra Cervone, Chandra Khatri, Rahul Goel, Behnam Hedayatnia, Anu Venkatesh, Dilek Hakkani-Tur, and Raefer Gabriel. 2019. Natural language generation at scale: A case study for open domain question answering. In *arXiv preprint arXiv:1903.08097*.

Yun-Nung Chen, Asli Celikyilmaz, and Dilek Hakkani-Tür. 2017. Deep learning for dialogue systems. *Proceedings of ACL 2017, Tutorial Abstracts*, pages 8–14.

Ondrej Dusek, Jekaterina Novikova, and Verena Rieser. 2017. Referenceless quality estimation for natural language generation. *CoRR*, abs/1708.01759.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Shi Feng, Eric Wallace, Pedro Grissom II, Alvin Rodriguez, Mohit Iyyer, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretation difficult. In *Empirical Methods in Natural Language Processing*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, page 94–104. Association for Computational Linguistics.

M Gašić, N Mrkšić, Pei-hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Policy committee for adaptation in multi-domain spoken dialogue systems. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 806–812. IEEE.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Sergey Golovanov, Rauf Kurbanov, Sergey Nikolenko, Kyryl Truskovskyi, Alexander Tselousov, and Thomas Wolf. 2019. Large-scale transfer learning for natural language generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6058.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*.

Vrindavan Harrison, Lena Reed, Shereen Oraby, and Marilyn Walker. 2019. Maximizing stylistic control and semantic accuracy in nlg: Personality variation and discourse contrast. In *Proceedings of the 1st Workshop on Discourse Structure in Neural NLG*, pages 1–12.

Vrindavan Harrison and Marilyn Walker. 2018. Neural generation of diverse questions using answer focus, contextual and linguistic features. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 296–306.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323.

Chris Kedzie and Kathleen McKeown. 2019. A good sample is hard to find: Noise injection sampling and self-training for neural language generation models. In *Proceedings of the 12th International Conference on Natural Language Generation*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Shikib Mehri and Maxine Eskenazi. 2020. Unsupervised evaluation of interactive dialog with dialogpt. In *Proc. of the SIGDIAL 2020*.

Neha Nayak, Dilek Hakkani-Tur, Marilyn Walker, and Larry Heck. 2017. To plan or not to plan? discourse planning in slot-value informed sequence to sequence models for language generation. In *Proc. of Interspeech 2017*.

J. Novikova, O. Dušek, and V. Rieser. 2017a. The e2e dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Conference*.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017b. Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252.

Shereen Oraby, Lena Reed, Shubhangi Tandon, TS Sharath, Stephanie Lukin, and Marilyn Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. In *SIGDIAL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Lena Reed, Shereen Oraby, and Marilyn Walker. 2018. Can neural generators for dialogue learn sentence planning and discouse structuring? In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 284–295.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page 35–40. Association for Computational Linguistics.

Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, volume 3, pages 41–51.

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically informative text generation. 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 4060–4067.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *Computational Linguistics and Intelligent Text Processing: 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19, 2005, Proceedings*, volume 3406, page 341. Springer.

Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentation in spoken dialogue systems. In *Meeting of the Association for Computational Linguistics*.

Amanda Stent, Marilyn Walker, Steve Whittaker, and Preetam Maloor. 2002. User-tailored generation for spoken dialogue: An experiment. In *ICSLP*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ye Tian, Ioannis Douratsos, and Isabel Groves. 2018. Treat the system like a human student: Automatic naturalness evaluation of generated text without reference texts. *INLG 2018*, page 109.

Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, et al. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 Conference on Empirical Methods*

in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2153–2162.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–129.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. *CoRR*, abs/1707.08052.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in japanese-to-english neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210.

A Supplementary Materials: Learning from Mistakes: Combining Ontologies via Self-Training for Dialogue Generation

A.1 Types of Semantic Errors

The TTM is tuned to identify 4 common neural generation errors: *deletions* (failing to realize a value), *repetitions* (repeating an attribute), *substitutions* (mentioning an attribute with an incorrect value), and *hallucinations* (introducing an attribute that was not in the original MR at all).

Table 9 illustrates each of these types of semantic errors. Row 1 shows deletions of *cuisine*, *price* and *near* which are in the MR but not in the realization. Row 2 demonstrates a repetition, where *location* and *decor* are both repeated. *Decor* is realized with two different lexical values, "good ambiance" and "good decor". There is a substitution in Row 3 where the MR states that the *food quality* is "bad", but *food quality* is realized as "good". Finally, Row 4 has a hallucination, *service* is not in the MR but it in the second sentence of the realization.

A.2 Example Errorful NLG Model Outputs

Table 10 provides examples of NLG model output utterances with high SERs. It illustrates how the NLG models struggle to combine attributes from the two ontologies which is required by all the input MRs (Column SB). It also illustrates cases where it is not possible to produce a valid retrofit MR that can be added back into training during self-training (Column Valid). In most cases these are due to many repetitions. Row 1 is an example where there is no source blending and since it has a repetition (*price*) it cannot be used for self-training (valid = no). Row 1 also illustrates an ungrammatical realization of *price* which we have no way to automatically detect at present *it is in the high price.* Row 2 has three deletions as well as two repetitions. The output repeats *It is in midtown* three times in a row. Row 3 has five errors, it does not realize the dialogue act RECOMMEND and has deleted three other attributes and it hallucinations *food quality*. While this is a significant number of errors, this realization can still be used in self-training, since none of its errors are repetitions. Row 4 has all four types of errors. It deletes *cuisine*, *decor* and *service*, it realizes a value for *family friendly* twice with different values, a substitution and finally it hallucinates *food quality*. Row 5 actually has more errors than slots. It deletes all but two of its attributes: *name* and *rating*. It also hallucinates *food quality* and repeats *rating*.

Model	Training	Test	SER	PERF %
BOOL	S-REPEAT	E2E	.14	25%
BOOL	S-REPEAT	NYC	.05	77%

Table 8: Performance of the self-trained S-Repeat model on the original E2E and NYC test sets. E2E test N = 630. NYC test N = 314.

A.3 Performance on E2E and NYC test sets

Table 2 provided a baseline for NOSUP's performance before self-training on the original test sets for E2E and NYC. We also verify that the self-trained model performs well after self-training. Table 8 shows that self-training improves the results for the original E2E and NYC test sets.

A.4 Example Final Model Outputs

Table 11 provides outputs from the final iteration of self-training that display different ways of combining different attributes from the ontologies. Row 1 shows that the model can combine attributes from the two sources in the same sentence, with attributes from each source, *decor* and *rating*, appearing in a single sentence with *and*. Row 2 shows a different way of combining attributes from the two sources, with *family friendly* and *food quality*, in a single sentence, this time using *with*. In Row 3 we can see that the model can also generate complex sentences for recommendations using the marker *because*. Also, the attribute used in the *because* clause is from E2E i.e. *family friendly* but such sentences never appear in the original E2E training data. The last row shows a complex sentence where *decor* is combined with *eat type* and *customer rating*, again a novel combination.

Error Type	MR	Realization
Delete CUISINE, PRICE, NEAR	name[RESTAURANT], cuisine[mexican], location[midtown], price[expensive], eatType[coffee shop], familyFriendly[no], near[POINT-OF-INTEREST]	[RESTAURANT] is a coffee shop that is not family friendly. It is located in Midtown.
Repeat LOCATION, DECOR	name[RESTAURANT], decor[good], location[midtown west], eatType[coffee shop], rating[1 out of 5]	[RESTAURANT] is a coffee shop in Midtown West with good ambiance. It is in Midtown West with good decor.
Substitution QUALITY BAD to GOOD	name[RESTAURANT], decor[good], qual[bad], location[tribeca/soho], eatType[pub]	[RESTAURANT] is in Tribeca/Soho with good food and good decor. It is a pub.
Hallucination SERVICE	name[RESTAURANT], decor[good], qual[good], location[riverside], near[POINT-OF-INTEREST]	[RESTAURANT] is near [POINT-OF-INTEREST] in the riverside area. It has good food, good decor and good service.

Table 9: Model outputs illustrating types of semantic errors.

#	SER	Valid	SB	MR	Realization
1	.60	no	no	name[RESTAURANT], `decor[fantastic]`, `qual[acceptable]`, price[high], *familyFriendly[yes]*	[RESTAURANT] is a child friendly restaurant with french food and it is in the high price range. it is in the high price.
2	.63	no	no	name[RESTAURANT], `recommend[yes]`, cuisine[japanese], `decor[bad]`, location[midtown], `service[fantastic]`, *rating[low]*, *near[*POINT-OF-INTEREST*]*	[RESTAURANT] is the best restaurant since it is a japanese restaurant with bad ambiance and it is in midtown. it is in midtown. it is in midtown.
3	.71	yes	yes	name[RESTAURANT], `recommend[yes]`, `decor[fantastic]`, `service[acceptable]`, *eatType[pub]*, *rating[high]*, *near[xnear]*	[RESTAURANT] is a pub with good food and it is located near the xnear.
4	.75	no	yes	name[RESTAURANT], cuisine[southern], `decor[fantastic]`, location[city centre], price[cheap], `service[good]`, *eatType[coffee shop]*, *familyFriendly[no]*	[RESTAURANT] is a cheap, family friendly coffee shop with good food. it is in the city centre. it is not family friendly.
5	1.17	no	yes	name[RESTAURANT], cuisine[indian], `decor[fantastic]`, `service[good]`, *rating[high]*, *near[*POINT-OF-INTEREST*]*	[RESTAURANT] provides good food and has great customer rating and has great customer rating .

Table 10: Example pathological outputs with high SERs from the NLG models before self-training. Valid realizations (col 3) are defined as those usable for self-training. In the MRs, the NYC attributes are represented using `typewriter font` and E2E attributes are represented using *italics*.

#	MR	Realization
1	name[RESTAURANT],cuisine[Indian], `decor[fantastic]`,`qual[good]`, `service[good]`, *rating[high]*, *near[*POINT-OF-INTEREST*]*	[RESTAURANT] is a Indian restaurant with great service and excellent ambiance and a high customer rating. It is located near the [POINT-OF-INTEREST].
2	name[RESTAURANT], `recommend[yes]`, `decor[good]`, `qual[good]`, price[cheap], `service[bad]`, *familyFriendly[yes]*,	[RESTAURANT] is a family friendly restaurant with good food, good ambiance and bad service. It is in the low price range.
3	name[RESTAURANT], `recommend[yes]`, `decor[bad]`, `qual[good]`, location[flatiron/union square], price[cheap], `service[acceptable]`, *eatType[coffee shop]*, *rating[3 out of 5]*,	[RESTAURANT] is the best restaurant because it is a family friendly coffee shop with good food, friendly service and bad ambiance. It is in Flatiron/Union Square. It has a customer rating of 3 out of 5. It is cheap.
4	name[RESTAURANT], `recommend[yes]`, cuisine[mediterranean], `decor[fantastic]`, price[very expensive], *eatType[pub]*, *rating[5 out of 5]* .	[RESTAURANT] is a Mediterranean pub with excellent ambiance and a customer rating of 5 out of 5. It is in the upper price range. It is the best restaurant.

Table 11: Example outputs of source blending from final self-training iterations. In the MRs, the NYC attributes are represented using `typewriter font` and E2E attributes are represented using *italics*.