Slug2Slug: A Deep Ensemble Model with Slot Alignment for Sequence-to-Sequence Natural Language Generation

Juraj Juraska, Panagiotis Karagiannis, Kevin K. Bowden and **Marilyn A. Walker**
Natural Language and Dialogue Systems Lab
University of California, Santa Cruz
{jjuraska,pkaragia,kkbowden,mawalker}@ucsc.edu

Abstract

Natural language generation lies at the core of generative dialogue systems and conversational agents. We describe an ensemble neural language generator, and present several novel methods for data representation and augmentation that yield improved results in our model. We test the model on three datasets in the restaurant, TV and laptop domains, and report both objective and subjective evaluations of our best model. Using a range of automatic metrics, as well as human evaluators, we show that our approach achieves better results than state-of-the-art models on the same datasets.

1 Introduction

There has recently been a substantial amount of research in natural language processing (NLP) in the context of personal assistants, such as Cortana or Alexa. The capabilities of these conversational agents are still fairly limited and lacking in various aspects, one of the most challenging of which is the ability to produce utterances with human-like coherence and naturalness for many different kinds of content. This is the responsibility of the natural language generation (NLG) component.

Our work focuses on language generators whose inputs are structured *meaning representations* (MRs). An MR describes a single dialogue act with a list of key concepts which need to be conveyed to the human user during the dialogue. Each piece of information is represented by a slot-value pair, where the *slot* identifies the type of information and the *value* is the corresponding content (see Table 3). *Dialogue act* (DA) types vary depending on the dialogue manager, ranging from simple ones, such as a *goodbye* DA with no slots at all, to complex ones, such as an *inform* DA containing multiple slots with various types of values.

A natural language generator must produce a syntactically and semantically correct utterance from a given MR. The utterance should express all the information contained in the MR, in a natural and conversational way. In traditional language generator architectures, the process of assembling an utterance from an MR is performed in two stages: *sentence planning*, which enforces semantic correctness and determines the structure of the utterance (order in which the information is presented, number of sentences, etc.), and *surface realization*, which enforces syntactic correctness and produces the final utterance form.

Earlier work on statistical NLG (SNLG) approaches were typically hybrids of a handcrafted component and a statistical training method. One of the first such generators, from Langkilde and Knight (1998), augmented a highly unconstrained rule-based surface realizer with statistical reranking based on an n-gram language model. Stent et al. (2004) took this idea further by applying it to sentence planning, and training the ranking function from human feedback. Other research on SNLG applied reinforcement learning to train a language generation policy (Rieser and Lemon, 2010). In all of these approaches, the handcrafted aspects lead to decreased portability and potentially limit the variability of the outputs. New corpus-based approaches emerged that used semantically aligned data to train language models that output utterances directly from their MRs (Mairesse et al., 2010; Mairesse and Young, 2014). The alignment provides valuable information during training, but the semantic annotation is costly.

The most recent methods do not require aligned data and use an *end-to-end* approach to training, performing sentence planning and surface realization simultaneously (Konstas and Lapata, 2013). The most successful systems trained on unaligned data use recurrent neural networks paired with an encoder-decoder system design (Wen et al., 2015; Mei et al., 2016; Dušek and Jurčíček, 2016), or

Table 1: Overview of the number of samples, as well as different DA and slot types, in each dataset.

	E2E	TV	Laptop
|training set|	42061	4221	7944
|validation set|	4672	1407	2649
|test set|	630	1407	2649
total	47363	7035	13242
DA types	1	14	14
slot types	8	16	20

other concepts, such as imitation learning (Lampouras and Vlachos, 2016). These language generation models, however, typically require greater amount of data for training, due to the lack of semantic alignment, and they still have problems producing syntactically and semantically correct output, as well as being limited in naturalness (Nayak et al., 2017).

Here we present a deep ensemble attentional encoder-decoder natural language generator, which we train and test on three large unaligned datasets in the restaurant, television, and laptop domains. We explore novel ways to represent the MR inputs, including novel methods for delexicalizing slots and their values, automatic slot alignment, as well as the use of a semantic reranker. We use automatic evaluation metrics to show that these novel methods appreciably improve the performance of our model. On the largest of the datasets, the E2E dataset (Novikova et al., 2017b) with 50K samples, we also demonstrate that our model significantly outperforms the baseline E2E NLG system in human evaluation. Finally, after augmenting our model with stylistic data selection, subjective evaluations reveal that it can still produce overall better results despite a significantly reduced training set.

2 Datasets

We evaluated the models on three datasets from different domains. The primary one is the recently released E2E restaurant dataset (Novikova et al., 2017b) with 48K samples. For benchmarking we use the TV dataset and the Laptop dataset (Wen et al., 2016) with 7K and 13K samples. Table 1 summarizes the proportions of the training, validation, and test sets for each dataset.

2.1 E2E Dataset

The E2E dataset is by far the largest one available for task-oriented language generation in the restaurant domain. The human references were collected using pictures as the source of information, which was shown to inspire more informative and natural utterances (Novikova et al., 2016). With nearly 50K samples, it offers almost 10 times more data than the San Francisco restaurant dataset introduced in Wen et al. (2015), which has frequently been used for benchmarks. The reference utterances in the E2E dataset exhibit superior lexical richness and syntactic variation, including more complex discourse phenomena. It aims to provide higher-quality training data for end-to-end language generation systems to learn to produce more naturally sounding utterances. The dataset was released as a part of the E2E NLG Challenge[1].

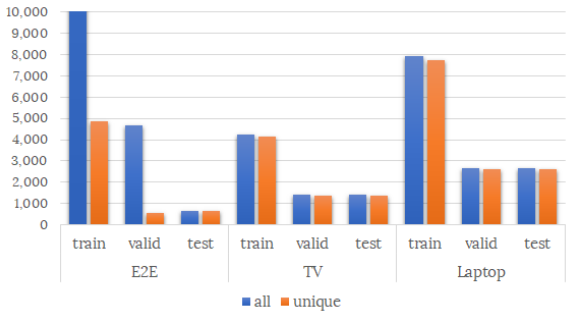

Figure 1: Proportion of unique MRs in the datasets. Note that the number of MRs in the E2E dataset was cut off at 10K for the sake of visibility of the small differences between other column pairs.

Although the E2E dataset contains a large number of samples, each MR is associated on average with 8.65 different reference utterances, effectively offering less than 5K unique MRs in the training set (Fig. 1). Explicitly providing the model with multiple ground truths, it offers multiple alternative utterance structures the model can learn to apply for the same type of MR. The delexicalization, as detailed later in Section 4.1, improves the ability of the model to share the concepts across different MRs.

The dataset contains only 8 different slot types, which are fairly equally distributed in the dataset. The number of slots in each MR ranges between 3 and 8, whereas the majority of MRs consist of 5 and 6 slots. Even though most of the MRs contain many slots, the majority of the corresponding human utterances, however, consist of one or two sentences only (Table 2), suggesting a reasonably high level of sentence complexity in the references.

[1] http://www.macs.hw.ac.uk/InteractionLab/E2E/

Table 2: Average number of sentences in the reference utterance for a given number of slots in the corresponding MR, along with the proportion of MRs with specific slot counts.

slots	3	4	5	6	7	8
sent.	1.09	1.23	1.41	1.65	1.84	1.92
prop.	5%	18%	32%	28%	14%	3%

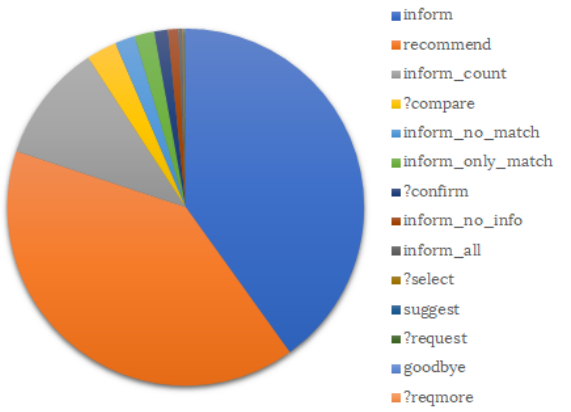

Figure 2: Proportion of DAs in the Laptop dataset.

2.2 TV and Laptop Datasets

The reference utterances in the TV and the Laptop datasets were collected using Amazon Mechanical Turk (AMT), one utterance per MR. Seeing only a single realization of each MR while training, the model must learn phrases and abstract constructs that it will need to apply to unseen MRs in order to perform well. Performing a delexicalization is thus even more essential here than in the E2E dataset. These two datasets are similar in structure, both using the same 14 DA types. The Laptop dataset, however, is almost twice as large and contains 25% more slot types. We noticed that the MRs with the `?request` DA type in the TV dataset have no slots provided, as opposed to the Laptop dataset, so we imputed these in order to obtain valid MRs.

Although both of these datasets contain more than a dozen different DA types, the vast majority (68% and 80% respectively) of the MRs describe a DA of either type `inform` or `recommend` (Fig. 2), which in most cases have very similarly structured realizations, comparable to those in the E2E dataset. DAs such as `suggest`, `?request`, or `goodbye` are represented by less than a dozen samples, but are significantly easier to learn to generate an utterance from because the corresponding MRs contain three slots at the most.

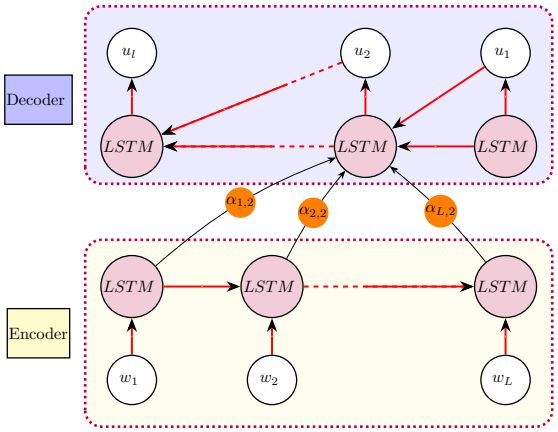

Figure 3: Standard architecture of a single-layer encoder-decoder LSTM model with attention. For each time step t in the output sequence, we calculate the attention scores $\alpha_{t,1}, \ldots, \alpha_{t,L}$. This diagram shows the attention scores only for $t = 2$.

3 Ensemble Neural Language Generator

3.1 Encoder-Decoder with Attention

For our sequence-to-sequence NLG model we use the standard *encoder-decoder* (Cho et al., 2014) architecture equipped with an *attention* mechanism as defined in Bahdanau et al. (2014). By encoding the input into a sequence of context vectors instead of a single vector, it enables the decoder to learn what specific parts of the input sequence to pay attention to, given the output generated so far.

In this attentional encoder-decoder architecture, the probability of the output at each time step t of the decoder depends on a distinct context vector q_t in the following way:

$$P(u_t|u_1, \ldots, u_{t-1}, \mathbf{w}) = g(u_{t-1}, s_t, q_t) \ ,$$

where in the place of function g we use the softmax function over the size of the vocabulary, and s_t is a hidden state of the decoder RNN at time step t, calculated as:

$$s_t = f(s_{t-1}, u_{t-1}, q_t) \ .$$

The context vector q_t is obtained as a weighted sum of all the hidden states $h_1, \ldots, h_L$ of the encoder:

$$q_t = \sum_{i=1}^{L} \alpha_{t,i} h_i \ ,$$

where $\alpha_{t,i}$ corresponds to the attention score the t-th word in the target sentence assigns to the i-th item in the input MR.

We compute the attention score $\alpha_{t,i}$ using a multi-layer perceptron (MLP) jointly trained with the entire system (Bahdanau et al., 2014). The encoder's and decoder's hidden states at time i and t, respectively, are concatenated and used as the input to the MLP, namely:

$$\alpha_{t,i} = softmax\left(\mathbf{w}^T tanh\left(W[h_i; s_t]\right)\right) ,$$

where W and $\mathbf{w}$ represent the weight matrix and the vector of the first and the second layer of the MLP, respectively.

In order to further improve the prediction performance of the decoder, during the testing we use beam search with the beam width equal to 10.

3.2 Ensembling

In order to enhance the quality of the predicted utterances, we create three attentional neural models with different encoder cell types. Two of the models contain a bidirectional LSTM encoder cell type, whereas the third model contains a CNN encoder cell type. We train these models individually for a different number of epochs and then combine their predictions.

Initially, we attempted to combine the predictions of the models by averaging the log-probability at every given time step and then selecting the word with the maximum log-probability. We noticed that the quality as well as the BLEU score of our utterances decreased significantly. We believe that this is due to the fact that different models learn different sentence structures and hence combining predictions at the probability level results in incoherent utterances.

Therefore, instead of combining the models at the log-probability level, we accumulate the ten most probable predicted utterances from each model type using beam search, and allow the reranker (see Section 3.4) to rank all candidate utterances based on the proportion of slots they realized. Finally, our system predicts the utterance that received the highest score.

3.3 Slot Alignment

Our training data is inherently unaligned, meaning our model isn't certain which sentence in a multi-sentence utterance contains a given slot, which limits the model's robustness. To accommodate this, we create a heuristic-based slot aligner which automatically preprocesses the data. Its primary goal is to align chunks of text from the reference utterances with an expected value from the MR. Applications of our slot aligner are described in subsequent sections and in Table 3.

In our task, we have a finite set of slot realizations which must be detected. Moreover, from our training data we can see that most slots are realized by inserting a specific set of phrases into an utterance. Using this insight we construct a gazetteer which primarily searches for overlapping content between the MR and each sentence in an utterance. We construct our gazetteer by associating all possible slot realizations with their appropriate slot type. We additionally augment the gazetteer using a small set of handcrafted rules which capture cases not easily encapsulated by the above process, for example, associating the `priceRange` slot with a chunk of text using currency symbols or relevant lexemes, such as "cheap" or "high end". While handcrafted, these rules are transferable across domains, as they target the slots, not the domains, and mostly serve to counteract the noise in the E2E dataset. Finally, we use WordNet (Fellbaum, 1998) to further augment the size of our gazetteer by accounting for synonyms and other semantic relationships, such as associating "pasta" with the `food[Italian]` slot.

Originally our model assumed that, if a given slot is present in the MR, it is also present in the utterance. After further inspecting our training data, we realized there was a noticeable percentage of slots which were never realized, or realized incorrectly. Therefore, our aligner was designed with respect to precision – if a slot is indicated by the original MR, but not observable to our aligner, we remove it from the MR.

3.4 Reranker

As discussed in Section 3.2, our model uses beam search to produce a pool of the most probable predictions for a given MR. While these results have a probability score provided by the model, we found that relying entirely on this score often results in the system picking a candidate which is objectively worse than a lower scoring utterance (i.e. one missing more slots and/or realizing slots incorrectly). We therefore augment that score by multiplying it by the following score which takes the slot alignment into consideration:

$$s_{\text{align}} = \frac{N}{(N_u + 1) \cdot (N_o + 1)} ,$$

where N is the number of all slots in the given MR, and N_u and N_o represent the number of unaligned slots (those not observed by our slot aligner) and over-generated slots (those which have been realized but were not present in the original MR), respectively.

4 Data Preprocessing

4.1 Delexicalization

We enhance the ability of our model to generalize the learned concepts to unseen MRs by delexicalizing the training data. Moreover, it reduces the amount of data required to train the model. We identify the categorical slots whose values always propagate verbatim to the utterance, and replace the corresponding values in the utterance with placeholder tokens. The placeholders are eventually replaced in the output utterance in post-processing by copying the values from the input MR. Examples of such slots would be `name` or `near` in the E2E dataset, and `screensize` or `processor` in the TV and the Laptop dataset.

Previous work identifies categorical slots as good delexicalization candidates that improve the performance of the model (Wen et al., 2015; Nayak et al., 2017). However, we chose not to delexicalize those categorical slots whose values can be expressed in alternative ways, such as "less than $20" and "cheap", or "on the riverside" and "by the river". Excluding these from delexicalization may lead to an increased number of incorrect realizations, but it encourages diversity of the model's outputs by giving it a freedom to choose among alternative ways of expressing a slot-value in different contexts. This, however, assumes that the training set contains a sufficient number of samples displaying these type of alternations so that the model can learn that certain phrases are synonymous. With its multiple human references for each MR, the E2E dataset has this property.

As Nayak et al. (2017) point out, delexicalization affects the sentence planning and the lexical choice around the delexicalized slot value. For example, the realization of the slot `food[`*`Italian`*`]` in the phrase "serves *Italian* food" is valid, while the realization of `food[`*`fast food`*`]` in "serves *fast food* food" is clearly undesired. Similarly, a naive delexicalization can result in "a Italian restaurant", whereas the article should be "an". Another problem with articles is singular versus plural nouns in the slot value. For example, the slot `accessories` in the TV dataset, can take on values such as "remote control", as well as "3D glasses", where only the former requires an article before the value.

We tackle this issue by defining different placeholder tokens for values requiring different treatment in the realization. For example, the value "Italian" of the `food` slot is replaced by `slot_vow_cuisine_food`, indicating that the value starts with a vowel and represents a cuisine, while "fast food" is replaced by `slot_con_food`, indicating that the value starts with a consonant and cannot be used as a term for cuisine. The model thus learns to generate "a" before `slot_con_food` and "an" before `slot_vow_cuisine_food` whenever relevant, as well as to avoid generating the word "food" after `food`-slot placeholders that do not contain the word "cuisine".

4.2 Data Expansion

Slot Permutation

In our initial experiments, we tried expanding the training set by permuting the slot ordering in the MRs as suggested in Nayak et al. (2017). From different slot orderings of every MR we sampled five random permutations (in addition to the original MR), and created new pseudo-samples with the same reference utterance. The training set thus increased six times in size.

Using such an augmented training set might add to the model's robustness, nevertheless it did not prove to be helpful with the E2E dataset. In this dataset, we observed the slot order to be fixed across all the MRs, both in the training and the test set. As a result, for the majority of the time, the model was training on MRs with slot orders it would never encounter in the test set, which ultimately led to a decreased performance in prediction on the test set.

Utterance/MR Splitting

Taking a more utterance-oriented approach, we augment the training set with single-sentence utterances paired with their corresponding MRs. These new pseudo-samples are generated by splitting the existing reference utterances into single sentences and using the slot aligner introduced in Section 3.3 to identify the slots that correspond to each sentence. The MRs of the new samples are created as the corresponding subsets of slots and, whenever the sentence contains the name (of the

Table 3: An example of the utterance/MR splitting.

MR	name [The Waterman], food [English], priceRange [cheap], customer rating [average], area [city centre], familyFriendly [yes]
Utt.	There is a family-friendly, cheap restaurant in the city centre, called The Waterman. It serves English food and has an average rating by customers.
New MR #1	name [The Waterman], priceRange [cheap], area [city centre], familyFriendly [yes], *position [outer]*
New MR #2	name [The Waterman], food [English], customer rating [average], *position [inner]*

restaurant/TV/etc.) or a pronoun referring to it (such as "it" or "its"), the `name` slot is included too. Finally, a new `position` slot is appended to every new MR, indicating whether it represents the first sentence or a subsequent sentence in the original utterance. An example of this splitting technique can be seen in Table 3. The training set almost doubled in size through this process.

Since the slot aligner works heuristically, not all utterances are successfully aligned with the MR. The vast majority of such cases is caused by reference utterances in the datasets having incorrect or entirely missing slot mentions. Nevertheless, there is only a small proportion of those, and we leave them in the training set with the unaligned slots removed from the MR, so as to avoid confusing the model when learning from such samples.

4.3 Sentence Planning via Data Selection

The quality of the training data inherently imposes an upper bound on the quality of the predictions of our model. Therefore, in order to bring our model to produce more sophisticated utterances, we experimented with filtering the training data to contain only the most natural sounding and structurally complex utterances for each MR.

We assess the complexity and naturalness of each utterance by the use of discourse phenomena, such as contrastive cues, subordinate clauses, aggregations, and the number of sentences used to convey all the information in the corresponding MR. We identify these in the utterance's parse-tree produced by the Stanford CoreNLP toolkit (Manning et al., 2014) by defining a set of rules for extracting the discourse phenomena.

5 Evaluation

Researchers have generally used both automatic and human evaluation for this NLG task. In our results we report the following standard automatic evaluation metrics: BLEU (Papineni et al., 2002), NIST (Przybocki et al., 2009), METEOR (Lavie and Agarwal, 2007), and ROUGE-L (Lin, 2004). For the E2E dataset experiments, we additionally report the results of the human evaluation carried out on the CrowdFlower platform as a part of the E2E NLG Challenge.

5.1 Experimental Setup

We have built our ensemble model using the seq2seq framework (Britz et al., 2017) for TensorFlow. Our individual LSTM models used a bidirectional LSTM encoder with 512 cells per layer, and the CNN models used a pooling encoder as described in Gehring et al. (2017). The decoder in all models was a 4-layer RNN decoder with 512 LSTM cells per layer and with attention. The hyperparameters were determined empirically through a series of experiments. After experimenting with different beam search parameters, we settled on the beam width of 10. Moreover, we employed the length normalization of the beams as defined in Wu et al. (2016), in order to encourage the decoder to favor longer sequences. The length penalty providing the best results on the E2E dataset was 0.6, whereas for the TV and Laptop datasets it was 0.9 and 1.0, respectively.

5.2 Experiments on the E2E Dataset

We start by evaluating our system on the E2E dataset. Since the reference utterances in the test set were kept secret for the E2E NLG Challenge, we carried out the metric evaluation using the validation set. This was necessary for us to be able to narrow down the models that perform well compared to the baseline model. The final model selection was done based on a human evaluation of the models' outputs on the test set.

5.2.1 Automatic Metric Evaluation

In the first experiment, we assess what effect the augmenting of the training set via utterance splitting has on the performance of different models. The results in Table 4 show that both the LSTM and the CNN models clearly benefit from additional pseudo-samples in the training set. This can likely be attributed to the model having access to more granular information about which parts of the utterance correspond to which slots in the MR. This may assist the model in sentence planning and building a stronger association between parts

Table 4: Automatic metric scores of different models tested on the E2E dataset, both unmodified ($\bar{\text{s}}$) and augmented (s) by the utterance splitting.

		BLEU	NIST	METEOR	ROUGE
LSTM	$\bar{\text{s}}$	0.6664	8.0150	0.4420	0.7062
	s	0.6930	8.4198	0.4379	0.7099
CNN	$\bar{\text{s}}$	0.6599	7.8520	0.4333	0.7018
	s	0.6760	8.0440	0.4448	0.7055

Table 5: Automatic metric scores of three different models and their ensemble, tested on the *validation set* of the E2E dataset. LSTM2 differs from LSTM1 in that it was trained longer.

	BLEU	NIST	METEOR	ROUGE
LSTM1	0.6661	8.1626	0.4644	0.7018
LSTM2	0.6493	7.9996	0.4649	0.6995
CNN	0.6636	7.9617	0.4700	0.7107
Ensem.	0.6576	8.0761	0.4675	0.7029

of the utterance and certain slots, such as that "it" is a substitute for the name.

Testing our ensembling approach reveals that pooling and reranking predictions from different models produces an ensemble model that is overall more robust than the individual models, when we take all four different metrics into consideration. The individual models fail to perform well in all the metrics at once, whereas the ensembling creates a new model that is more consistent across the different metric types (Table 5). When evaluated on the official E2E test set, our ensemble model performs comparably to the baseline model, TGen (Dušek and Jurčíček, 2016), in terms of automatic metrics (Table 6).

5.2.2 Human Evaluation

It is known that automatic metrics function only as a general and vague indication of the quality of an utterance in a dialogue (Liu et al., 2016; Novikova et al., 2017a). Systems which score similarly according to these metrics could produce utterances that are significantly different because automatic metrics fail to capture many of the characteristics of natural sounding utterances. Therefore, to better assess the structural complexity of the predictions of our model, we present the results of a human evaluation of the models' outputs in terms of both naturalness and quality, carried out by the E2E NLG Challenge organizers.

Table 6: Automatic metric scores of our ensemble model compared against TGen (the baseline model), tested on the *test set* of the E2E dataset.

	BLEU	NIST	METEOR	ROUGE
TGen	0.6593	8.6094	*0.4483*	*0.6850*
Ensem.	*0.6619*	*8.6130*	0.4454	0.6772

Quality examines the grammatical correctness and adequacy of an utterance given an MR, whereas *naturalness* assesses whether a predicted utterance could have been produced by a native speaker, disregarding the corresponding MR. In order to obtain these scores, the crowd workers were presented with the outputs of five randomly selected systems, which they ranked from the best to worst. The systems' final scores were then produced using the TrueSkill algorithm (Sakaguchi et al., 2014) performing pairwise comparisons of the human evaluation scores among the twenty competing systems.

Our system, trained on the E2E dataset without the stylistic data selection (Section 4.3), achieved the highest quality score in the E2E NLG Challenge, and was ranked second in naturalness[2]. The system's performance in quality (the primary metric) was statistically significantly better than the competition according to the TrueSkill evaluation, which used bootstrap resampling with a p-level of $p \leq 0.05$.

Comparing these results with the scores achieved by the baseline model in quality and naturalness (5th and 6th place, respectively) reinforces our belief that models performing similarly according to automatic metrics (Table 6) can have vast differences in the structural complexity of their generated utterances.

5.2.3 Experiments with Data Selection

After reducing the E2E training set as described in Section 4.3, the new training set consisted of approximately 20K pairs of MRs and utterances. Despite this drastic reduction in training samples, the model was able to learn more complex utterances that contained the natural variations of the human language, such as aggregations or contrastive sentences using "but". Nevertheless, the model also failed to realize slots more frequently.

In order to observe the effect of stylistic data se-

[2]Note that the system that surpassed ours in naturalness was ranked the last according to the quality metric.

Table 7: Average error rate and naturalness metrics obtained from six annotators for different ensemble models.

Ensemble model	Error rate	Naturalness
Conservative	*0.40%*	2.196
Progressive	1.60%	2.118
Hybrid	*0.40%*	*2.435*

lection, we conducted a human evaluation where we assessed the utterances based on *error rate* and *naturalness*. The error rate is calculated as the percentage of slots the model failed to realize divided by the total number of slots present among all samples. The annotators ranked samples of utterance triples – corresponding to three different ensemble models – by naturalness from 1 to 3 (3 being the most natural, with possible ties). The *conservative* model combines three submodels all trained on the full dataset, the *progressive* one combines submodels solely trained on the filtered dataset, and finally, the *hybrid* is an ensemble of three models only one of which is trained on the full dataset, so as to serve as a fallback.

The impact of the reduction of the number of training samples becomes evident by looking at the score of the progressive model (Table 7), where this model trained solely on the reduced dataset had the highest error rate. We observe, however, that a hybrid ensemble model manages to perform the best in terms of the error rate, as well as the naturalness. These results suggest that filtering the dataset can actually help to achieve better and more natural sounding utterances.

5.3 Experiments on TV and Laptop Datasets

In order to provide a better frame of reference for the performance of our proposed model, we utilize the RNNLG benchmark toolkit[3] to evaluate our system on two additional, widely used datasets in NLG, and compare our results with those of a state-of-the-art model, SCLSTM (Wen et al., 2015). As Table 8 shows, our ensemble model performs competitively with the baseline on the TV dataset, and it outperforms it on the Laptop dataset by a wide margin. We believe the higher error rate of our model can be explained by the significantly less aggressive slot delexicalization than the one used in SCLSTM. That, however, gives our model a greater lexical freedom and, with it, the ability to produce more natural sounding utterances.

[3]https://github.com/shawnwun/RNNLG

Table 8: Automatic metric scores of our ensemble model evaluated on the test sets of the TV and Laptop datasets, and compared against SCLSTM. The ERR column indicates the slot error rate, as computed by the RNNLG toolkit (for our models calculated in post-processing).

	TV		Laptop	
	BLEU	**ERR**	**BLEU**	**ERR**
SCLSTM	0.5265	2.31%	0.5116	0.79%
LSTM	0.5012	3.86%	0.5083	4.43%
CNN	0.5287	1.87%	0.5231	2.25%
Ensem.	0.5226	1.67%	0.5238	1.55%

The model trained on the Laptop dataset is also a prime example of how an ensemble model is capable of extracting the best learned concepts from each individual submodel. By combining their knowledge and compensating thus for each other's weaknesses, the ensemble model can achieve a lower error rate, as well as a better overall quality, than any of the submodels individually.

6 Conclusion and Future Work

In this paper we have presented our ensemble attentional encoder-decoder NLG which is capable of generating natural utterances from MRs. Moreover we have also presented novel methods of representing the MRs to improve performance. Our results indicate that the proposed ensembling method, as well as the experimental stylistic data selection, greatly improve our model's ability to generalize and produce more natural sounding utterances, while minimizing the number of slots that are not realized during the generation.

In this paper we have seen that automatic slot alignment can expand our training data and rerank utterances. Our slot alignment currently relies in part on empirically observed heuristics, a more robust aligner would allow for more flexible expansion into new domains. Since the stylistic data selection noticeably improved the diversity of our system's outputs, we believe this is a method with future potential. Finally, it is clear that current automatic evaluation metrics are only sufficient at providing a vague idea as to the systems performance; we postulate that leveraging the reference data to train a classifier will result in a more conclusive automatic evaluation metric.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *ICLR* .

Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906* .

Merrienboer Cho, Bougares Gulcehre, and Bengio Schwenk. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP* .

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *ACL* .

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. 2017. A convolutional encoder model for neural machine translation. *ICLR* .

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *J. Artif. Intell. Res.(JAIR)* 48:305–346.

Gerasimos Lampouras and Andreas Vlachos. 2016. Imitation learning for language generation from unaligned data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, pages 1101–1112.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 704–710.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Stroudsburg, PA, USA, StatMT '07, pages 228–231. http://dl.acm.org/citation.cfm?id=1626355.1626389.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. http://www.aclweb.org/anthology/W04-1013.

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 *EMNLP*.

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1552–1561.

François Mairesse and Steve Young. 2014. Stochastic language generation in dialogue using factored language models. *Computational Linguistics* .

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. pages 55–60.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2016. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *NAACL* .

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.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017a. Why we need new evaluation metrics for nlg. In *EMNLP*.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017b. The E2E NLG shared task .

Jekaterina Novikova, Oliver Lemon, and Verena Rieser. 2016. Crowd-sourcing nlg data: Pictures elicit better data. *INLG* .

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '02, pages 311–318. https://doi.org/10.3115/1073083.1073135.

Mark Przybocki, Kay Peterson, Sébastien Bronsart, and Gregory Sanders. 2009. The nist 2008 metrics for machine translation challenge—overview, methodology, metrics, and results. *Machine Translation* 23(2):71–103. https://doi.org/10.1007/s10590-009-9065-6.

Verena Rieser and Oliver Lemon. 2010. Natural language generation as planning under uncertainty for spoken dialogue systems. In *Empirical methods in natural language generation*, Springer, pages 105–120.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical*

Machine Translation. Association for Computational Linguistics, Baltimore, Maryland, USA, pages 1–11. http://www.aclweb.org/anthology/W14-3301.

Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proceedings of the 42nd annual meeting on association for computational linguistics*. Association for Computational Linguistics, page 79.

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 on North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, 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 (EMNLP)*. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR* .