# 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. *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 (see example in Table 1).

| | |
|---|---|
| **MR** | *inform* (name [**The Golden Curry**], food [**Japanese**], priceRange [**moderate**], familyFriendly [**yes**], near [**The Bakers**]) |
| **Utt.** | Located near **The Bakers**, **kid-friendly** restaurant, **The Golden Curry**, offers **Japanese** cuisine with a **moderate** price range. |

Table 1: An example of an MR and a corresponding reference utterance.

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 assembling of an utterance from an MR is performed in two stages: *sentence planning*, which enforces semantic correctness and determines the structure of the utterance, and *surface realization*, which enforces syntactic correctness and produces the final utterance form.

Earlier work on statistical NLG approaches were typically hybrids of a handcrafted component and a statistical training method (Langkilde and Knight, 1998; Stent et al., 2004; Rieser and Lemon, 2010). The handcrafted aspects, however, 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 (RNNs) paired with an encoder-decoder system design (Mei et al.,

*Proceedings of NAACL-HLT 2018*, pages 152–162
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

2016; Dušek and Jurčíček, 2016), but also other concepts, such as imitation learning (Lampouras and Vlachos, 2016). These NLG 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 neural ensemble 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 methods appreciably improve the performance of our model. On the largest of the datasets, the E2E dataset (Novikova et al., 2017b) with nearly 50K samples, we also demonstrate that our model significantly outperforms the baseline *E2E NLG Challenge*[1] 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 Related Work

NLG is closely related to machine translation and has similarly benefited from recent rapid development of deep learning methods. State-of-the-art NLG systems build thus on deep neural *sequence-to-sequence* models (Sutskever et al., 2014) with an *encoder-decoder* architecture (Cho et al., 2014) equipped with an *attention* mechanism (Bahdanau et al., 2015). They typically also rely on slot *delexicalization* (Mairesse et al., 2010; Henderson et al., 2014), which allows the model to better generalize to unseen inputs, as exemplified by TGen (Dušek and Jurčíček, 2016). However, Nayak et al. (2017) point out that there are frequent scenarios where delexicalization behaves inadequately (see Section 5.1 for more details), and Agarwal and Dymetman (2017) show that a *character-level* approach to NLG may avoid the need for delexicalization, at the potential cost of making more semantic omission errors.

The end-to-end approach to NLG typically requires a mechanism for aligning slots on the output utterances: this allows the model to generate utterances with fewer missing or redundant slots. Cuayáhuitl et al. (2014) perform automatic slot labeling using a Bayesian network trained on a labeled dataset, and show that a method using spectral clustering can be extended to unlabeled data with high accuracy. In one of the first successful neural approaches to language generation, Wen et al. (2015a) augment the generator's inputs with a control vector indicating which slots still need to be realized at each step. Wen et al. (2015b) take the idea further by embedding a new sigmoid gate into their LSTM cells, which directly conditions the generator on the DA. More recently, Dušek and Jurčíček (2016) supplement their encoder-decoder model with a trainable classifier which they use to rerank the beam search candidates based on missing and redundant slot mentions.

| | 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 |

Table 2: Overview of the number of samples, as well as different DA and slot types, in each dataset .

Our work builds upon the successful attentional encoder-decoder framework for sequence-to-sequence learning and expands it through ensembling. We explore the feasibility of a domain-independent slot aligner that could be applied to any dataset, regardless of its size, and beyond the reranking task. We also tackle some challenges caused by delexicalization in order to improve the quality of surface realizations, while retaining the ability of the neural model to generalize.

## 3 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, respectively. Table 2 summarizes the proportions of the training, validation, and test sets for each dataset.

### 3.1 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

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

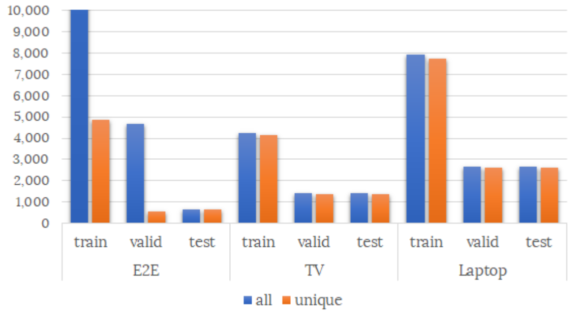

Figure 1: Proportion of unique MRs in the datasets. Note that the number of MRs in the E2E dataset was cut off at 10K for the sake of visibility of the small differences between other column pairs.

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. (2015b), 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 NLG systems to learn to produce more naturally sounding utterances. The dataset was released as a part of the E2E NLG Challenge.

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 5.1, 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. The number of slots in each MR ranges between 3 and 8, but the majority of MRs consist of 5 or 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 3), suggesting a reasonably high level of sentence complexity in the references.

### 3.2 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. These two datasets are similar in structure, both using the same 14 DA types.[2] The Laptop dataset, however, is almost twice as large and contains 25% more slot types.

| **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% |

Table 3: 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.

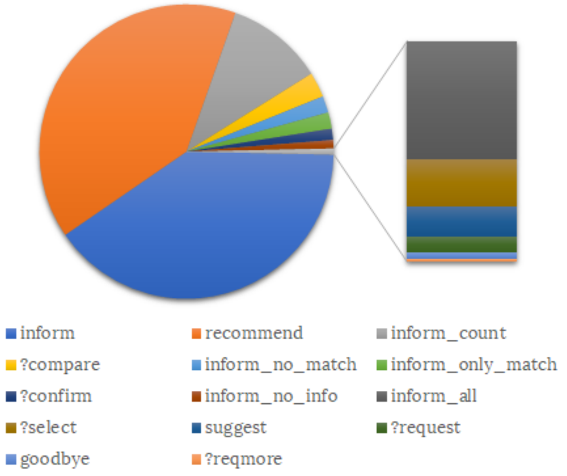

Figure 2: Proportion of DAs in the Laptop dataset.

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.

## 4 Ensemble Neural Language Generator

### 4.1 Encoder-Decoder with Attention

Our model uses the standard encoder-decoder architecture with attention, as defined in Bahdanau et al. (2015). Encoding the input into a sequence of context vectors instead of a single vector enables the decoder to learn what specific parts of the

[2] We noticed 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.

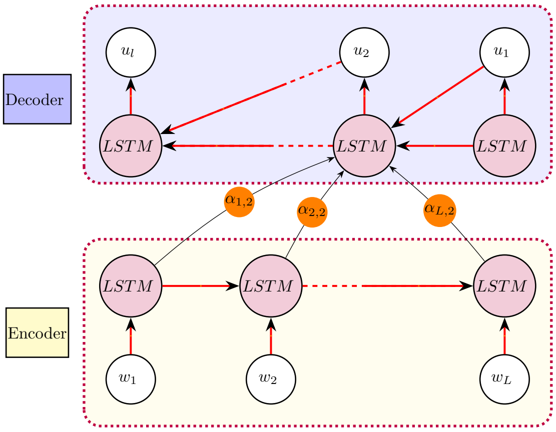

Figure 3: Standard architecture of a single-layer encoder-decoder LSTM model with attention. For each time step $t$ in the output sequence, the attention scores $\alpha_{t,1}, \ldots, \alpha_{t,L}$ are calculated. This diagram shows the attention scores only for $t = 2$.

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., 2015). 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 $\mathbf{w}$ are the weight matrix and the vector of the first and the second layer of the MLP, respectively. The learned weights indicate the level of influence of the individual words in the input sequence on the prediction of the word at time step $t$ of the decoder. The model thus learns a soft alignment between the source and the target sequence.

### 4.2 Ensembling

In order to enhance the quality of the predicted utterances, we create three neural models with different encoders. Two of the models use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) encoder, whereas the third model has a CNN (LeCun et al., 1998) encoder. 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 each 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 top 10 predicted utterances from each model type using beam search and allow the reranker (see Section 4.4) to rank all candidate utterances taking the proportion of slots they successfully realized into consideration. Finally, our system predicts the utterance that received the highest score.

### 4.3 Slot Alignment

Our training data is inherently unaligned, meaning our model is not 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 4.

In our task, we have a finite set of slot mentions which must be detected in the corresponding utterance. 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, 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.

### 4.4 Reranker

As discussed in Section 4.2, our model uses beam search to produce a pool of the most likely utterances 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.

## 5 Data Preprocessing

### 5.1 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., 2015b; 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 this type of alternation 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 instance, 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 when appropriate, as well as to avoid generating the word "food" after food-slot placeholders that do not contain the word "cuisine". All these rules are general and can automatically be applied across different slots and domains.

## 5.2 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 4.3 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 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 4. 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, however, is caused by reference utterances in the datasets having incorrect or entirely missing slot mentions. There is a noticeable proportion of those, so 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.

| | |
|---|---|
| **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]* |

Table 4: An example of the utterance/MR splitting.

| | |
|---|---|
| **MR** | name [Wildwood], eatType [coffee shop], food [English], priceRange [moderate], customer rating [1 out of 5], near [Ranch] |
| **Simple utt.** | Wildwood provides English food for a moderate price. It has a low customer rating and is located near Ranch. It is a coffee shop. |
| **Elegant utt.** | A low-rated English style coffee shop around Ranch, called Wildwood, has moderately priced food. |

Table 5: Contrastive example of a simple and a more elegant reference utterance style for the same MR in the E2E dataset.

## 5.3 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. For instance, we prefer having an elegant, single-sentence utterance with an apposition as the reference for an MR, rather than an utterance composed of three simple sentences, two of which begin with "it" (see the examples in Table 5).

We assess the complexity and naturalness of each utterance by the use of discourse phenomena, such as contrastive cues, subordinate clauses, or aggregation. 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. Furthermore, we consider the number of sentences used to convey all the information in the corresponding MR, as longer sentences tend to exhibit more advanced discourse phenomena. Penalizing utterances for too many sentences contributes to reducing the proportion of generic reference utterances, such as the "simple" example in the above table, in the filtered training set.

## 6 Evaluation

Researchers in NLG have generally used both automatic and human evaluation. Our results report the 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.

### 6.1 Experimental Setup

We built our ensemble model using the seq2seq framework (Britz et al., 2017) for TensorFlow. Our individual LSTM models use a bidirectional LSTM encoder with 512 cells per layer, and the CNN models use a pooling encoder as 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. 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.

### 6.2 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 to narrow down the models that perform well compared to the baseline. The final model selection was done based on a human evaluation of the models' outputs on the test set.

#### 6.2.1 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 6 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 of the utterance and certain slots, such as that "it" is a substitute for the name.

| | | BLEU | NIST | METEOR | ROUGE |
|---|---|---|---|---|---|
| **LSTM** | $\overline{\text{s}}$ | 0.6664 | 8.0150 | 0.4420 | 0.7062 |
| | s | 0.6930$^{\ddagger}$ | 8.4198 | 0.4379 | 0.7099 |
| **CNN** | $\overline{\text{s}}$ | 0.6599 | 7.8520 | 0.4333 | 0.7018 |
| | s | 0.6760$^{\dagger}$ | 8.0440 | 0.4448 | 0.7055 |

Table 6: Automatic metric scores of different models tested on the E2E dataset, both unmodified ($\overline{\text{s}}$) and augmented (s) through the utterance splitting. The symbols $^{\dagger}$ and $^{\ddagger}$ indicate statistically significant improvement over the $\overline{\text{s}}$ counterpart with $p < 0.05$ and $p < 0.01$, respectively, based on the paired t-test.

Testing our ensembling approach reveals that reranking predictions pooled from different models produces an ensemble model that is overall more robust than the individual submodels. The submodels fail to perform well in all four metrics at once, whereas the ensembling creates a new model that is more consistent across the different metric types (Table 7).[3] While the ensemble model decreases the proportion of incorrectly realized slots compared to its individual submodels on the validation set, on the test set it only outperforms two of the submodels in this aspect (Table 8). Analyzing the outputs, we also observed that the CNN model surpassed the two LSTM models in the ability to realize the "fast food" and "pub" values reliably, both of which were hardly present in the validation set but very frequent in the test set. 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 9).

#### 6.2.2 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

[3] The scores here correspond to the model submitted to the E2E NLG Challenge. Subsequently, we found better performing models according to some metrics: see Table 6.

| | 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 |

Table 7: 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.

| | Validation set | Test set |
|---|---|---|
| **LSTM1** | 0.116% | 0.988% |
| **LSTM2** | 0.145% | 1.241% |
| **CNN** | 0.232% | 0.253% |
| **Ensem.** | 0.087% | 0.965% |

Table 8: Error rate of the ensemble model compared to its individual submodels.

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

Table 9: Automatic metric scores of our ensemble model compared against TGen (the baseline model), tested on the *test set* of the E2E dataset.

| | |
|---|---|
| **Ex. #1** | The Cricketers is a cheap Chinese restaurant near All Bar One in the riverside area, **but it has an average customer rating and is not family friendly**. |
| **Ex. #2** | **If you are looking for** a coffee shop near The Rice Boat, **try** Giraffe. |

Table 10: Examples of generated utterances that contain more advanced discourse phenomena.

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.

*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, irrespective of the MR. To obtain these scores, crowd workers ranked the outputs of 5 randomly selected systems from worst to best. The final scores were produced using the TrueSkill algorithm (Sakaguchi et al., 2014) through pairwise comparisons of the human evaluation scores among the 20 competing systems.

Our system, trained on the E2E dataset without stylistic selection (Section 5.3), achieved the highest quality score in the E2E NLG Challenge, and was ranked second in naturalness.[4] The system's performance in quality (the primary metric) was 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 that perform similarly on the automatic metrics (Table 9) can exhibit vast differences in the structural complexity of their generated utterances.

#### 6.2.3 Experiments with Data Selection

After filtering the E2E training set as described in Section 5.3, the new training set consisted of approximately 20K pairs of MRs and utterances. Interestingly, 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. The generated utterances exhibited discourse phenomena such as contrastive cues (see Example #1 in Table 10), as well as a more conversational style (Example #2). Nevertheless, the model also failed to realize slots more frequently.

In order to observe the effect of stylistic data selection, 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 training set, 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 training set, so as to serve as a fallback.

The impact of the reduction of the number of

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

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

Table 11: Average error rate and naturalness metrics obtained from six annotators for different ensemble models.

training samples becomes evident by looking at the score of the progressive model (Table 11), 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 through careful data selection can help to achieve better and more natural sounding utterances. It significantly improves the model's ability to produce more elegant utterances beyond the "[name] is... It is/has..." format, which is only too common in neural language generators in this domain.

### 6.3 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[5] 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., 2015b). As Table 12 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 utterances.

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.

| | **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% |

Table 12: 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).

## 7 Conclusion and Future Work

In this paper we presented our ensemble attentional encoder-decoder model for generating natural utterances from MRs. Moreover, we presented novel methods of representing the MRs to improve performance. Our results indicate that the proposed utterance splitting applied to the training set greatly improves the neural model's accuracy and ability to generalize. The ensembling method paired with the reranking based on slot alignment also contributed to the increase in quality of the generated utterances, while minimizing the number of slots that are not realized during the generation. This also enables the use of a less aggressive delexicalization, which in turn stimulates diversity in the produced utterances.

We showed that automatic slot alignment can be utilized for expanding the training data, as well as for utterance reranking. Our alignment currently relies in part on empirically observed heuristics, and 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, which we intend to further explore. Finally, it is clear that current automatic evaluation metrics in NLG are only sufficient for providing a vague idea as to the system's performance; we postulate that leveraging the reference data to train a classifier will result in a more conclusive automatic evaluation metric.

## Acknowledgements

This research was partially supported by NSF Robust Intelligence #IIS-1302668-002.

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

# References

Shubham Agarwal and Marc Dymetman. 2017. A surprisingly effective out-of-the-box char2char model on the e2e nlg challenge dataset. In *SIGDIAL Conference*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR* .

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc V. Le. 2017. Massive exploration of neural machine translation architectures. *CoRR* abs/1703.03906.

Kyunghyun Cho, Bart van Merriënboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Heriberto Cuayáhuitl, Nina Dethlefs, Helen Hastie, and Xingkun Liu. 2014. Training a statistical surface realiser from automatic slot labelling. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, pages 112–117.

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

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

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, pages 360–365.

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

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 *COLING*.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING-ACL*.

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, pages 228–231.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* .

Chia-Wei Liu, Ryan Joseph Lowe, Iulian 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 *ACL*.

François Mairesse and Steve Young. 2014. Stochastic language generation in dialogue using factored language models. *Computational Linguistics* 40:763–799.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL*.

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. In *INTERSPEECH*.

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 *ACL*.

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(2-3):71–103.

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, pages 1–11.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić, Pei hao Su, David Vandyke, and Steve Young. 2015a. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *SIGDIAL Conference*.

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 *NAACL*.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015b. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *EMNLP*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR* abs/1609.08144.