INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

To Plan or not to Plan? Discourse planning in slot-value informed sequence to sequence models for language generation

Neha Nayak[1], *Dilek Hakkani-Tür*[1], *Marilyn Walker*[1,2], and *Larry Heck*[1]

[1]Google Research, Mountain View, Ca, U.S.A.
[2]University of California Santa Cruz, U.S.A.

nayakn@google.com, dilek@ieee.org, mawalker@ucsc.edu, larry.heck@ieee.org

Abstract

Natural language generation for task-oriented dialogue systems aims to effectively realize system dialogue actions. All natural language generators (NLGs) must realize grammatical, natural and appropriate output, but in addition, generators for task-oriented dialogue must faithfully perform a specific dialogue act that conveys specific semantic information, as dictated by the dialogue policy of the system dialogue manager. Most previous work on deep learning methods for task-oriented NLG assumes that generation output can be an utterance skeleton. Utterances are delexicalized, with variable names for slots, which are then replaced with actual values as part of post-processing. However, the value of slots do, in fact, influence the lexical selection in the surrounding context as well as the overall sentence plan. To model this effect, we investigate sequence-to-sequence (seq2seq) models in which slot values are included as part of the input sequence and the output surface form. Furthermore, we study whether a separate sentence planning module that decides on grouping of slot value mentions as input to the seq2seq model results in more natural sentences than a seq2seq model that aims to jointly learn the plan and the surface realization.
Index Terms: generation, dialog, deep learning, sentence planning

1. Introduction

Natural language generation for task-oriented dialogue systems aims to effectively realize system dialogue actions. All natural language generators (NLGs) have significant, complex constraints on the grammaticality, naturalness and appropriateness of their output, but, unlike chat-based systems, the NLG in task-oriented dialogue must faithfully perform a specific dialogue act that conveys *specific semantic information*, as dictated by the dialogue manager's (DM's) policy [1, 2]. Table 1 shows the semantics of dialogue acts that the DM might specify in the restaurant domain (in gray), while Rows 1 and 2, and Rows 4 and 5 illustrate how a single semantic specification from the DM can result in several different outputs from the NLG.

In order to train the NLG for task-oriented dialogue, it is thus necessary to provide training data that represents the mapping from semantics to surface realizations[3]. To date, this has meant that the data collection must target the particular task, whereas NLG for chatbots can often rely on large-scale, harvested, social media dialogues [4, 5, 6, 7]. The result is that datasets for training NLGs for task-oriented dialogue are typically smaller in scale than those used for chatbots [8, 9, 10, 11, 12, 13].

To combat the effects of data sparsity, a technique that is commonly used in task-oriented generation is that of delexicalization [14, 15]. In the input to NLG, the mentions in each utterance are described in terms of slots and values. In delexicalization, slot values occurring in training utterances are replaced with a placeholder token representing the slot, such that weights can be shared between similar utterances. For example, Rows 6 and 7 of Table 2 are delexicalized versions of Rows 1 and 2 of Table 1. At generation time, the NLG generates outputs with these placeholders, which are then filled in from a semantic frame, or copied over from the input specification to form the final output [15, 14]. This is especially beneficial in the presence of 'open-vocabulary' slots such as restaurant names, which can take on values from a virtually unlimited set.

Table 1: *NLG for Restaurants I*

#	`recommend(restaurant_name= Au Midi, neighborhood = midtown, cuisine = french`
1	Au Midi is in Midtown and serves French food.
2	There is a French restaurant in Midtown called Au Midi.
	`recommend(restaurant_name= Emilio's, neighborhood = Brooklyn, cuisine = italian)`
3	There is **an** Italian restaurant in Brooklyn called Emilio's.
	`recommend(restaurant_name= Loch Fyne, neighborhood = city centre, cuisine = seafood)`
4	Loch Fyne is in **the** City Center and serves seafood food.
5	There is a seafood restaurant in the City Centre called Loch Fyne.

Table 2: *Delexicalized Representations*

#	`recommend(restaurant_name= Au Midi, neighborhood = midtown, cuisine = french)`
6	`restaurant_name` is in `neighborhood` and serves `cuisine` food.
7	There is a `cuisine` restaurant in `neighborhood` called `restaurant_name`.

Thus, in cases where it is valid to assume that the slot values appear verbatim in the utterances, and the surrounding tokens are independent of the slot values, delexicalization is an effective strategy for reducing the amount of training data needed. This assumption is, however, violated in many cases that occur even in simple task-oriented NLG scenarios. Some slots are *nondelexicalizable* [14]. Moreover, slot values **do** influence the structure of an utterance, as well as localized decisions such as the choice of tokens surrounding the mention of the slot in the utterance, and aspects of syntactic choice. Production of this underlying structure in the traditional NLG architecture is called *sentence planning*[16, 17].

Recently, deep learning methods, such as Semantically Conditioned LSTM (SC-LSTM) and sequence-to-sequence (seq2seq) models have been shown to be effective for NLG in task-oriented dialogue [14, 18, 19, 20]. However, deep learning methods have not attempted to replicate sentence plan-

http://dx.doi.org/10.21437/Interspeech.2017-1525

ning phenomena, such as aggregation, lexical choice, discourse structuring, and context sensitivity, that were often the focus of previous statistical language generation approaches [21, 22, 23, 24, 25, 26, 27]. While deep learning models theoretically have the capacity to learn such higher-level semantically-driven phenomena from data, in practice such behavior tends to require large amounts of data that are prohibitively expensive for task-oriented NLG.

In this paper, we present a dataset selected to encourage aggregation and discourse structuring. We work with seq2seq models that take as input slot values for scalar slot categories as part of the input sequence. We compare the lexicalized outputs from such models with previous approaches that output delexicalized sequences, and then generate the final output with post processing to insert slot values. To study the effect of sentence planning, we incrementally introduce supervision for sentence planning phenomena as input to the model.

2. Motivation

Slot values affect the utterance skeleton in NLG in several ways. The scalar valued slots in our restaurant domain schema (Table 3) introduce additional opportunities for variation.

Table 3: *Restaurant Entity Schema. Example values for categorical slots are shown. All scalar valued slots share the range of four possible values.*

	Slot	Possible value
categorical	Name	Au Midi
	Neigborhood	Midtown
	Cuisine	French
scalar	Decor	excellent
	Food quality	good
	Service	decent
	Value for money	mediocre

- Slot values impact lexical choice. For example, `french` as a cuisine type is realized as *French food* in Row 1 of Table 1, but `seafood` as a cuisine type cannot be realized as *seafood food*: see Row 4. However when the `cuisine` slot is in a different syntactic position, e.g. *seafood restaurant* in Row 5, a special rule about the `seafood` value is not needed. Compare Row 5 with Row 2. Determiners also depend on the slot-value, such as the difference in determiner between `midtown` and `city centre` in Rows 2 and 5.
- When an utterance realizes multiple slots, the most natural utterance skeleton often depends on the values of the slots. If the slot values are the same, humans typically group them together in a single clause, as in Row 8 of Table 4. If the values are different, contrastive discourse connectives may be used as in Row 9 in Table 4.

Thus, delexicalized representations lose important information that could be used for sentence planning. This has made it impossible to directly address the issues described above in previous work on seq2seq models. In the following section, we describe extensions to a standard seq2seq generation model. We attempt to learn to group mentions of similar slots together (slot aggregation), and to structure sentences with discourse markers (discourse structuring).

3. Neural Models for NLG

We first describe the sequence-to-sequence model that is the basis for most neural NLG systems, and then some extensions that demonstrate the need for sentence planning.

Table 4: *NLG for Restaurants II*

	`recommend(restaurant_name= Loch Fyne, food_quality = excellent, cuisine = seafood, decor = excellent, service = excellent)`
8	Loch Fyne is a pretty solid restaurant. They specialize in seafood cuisine, and people report the decor, service, and food quality are **all** excellent.
	`recommend(restaurant_name= Emilio's, food_quality = excellent, location = city centre, cuisine = italian, decor = decent, service = decent)`
9	Emilio's decor and service are **both** decent, **but its** food quality is nothing short of excellent. It serves Italian food and its in the City Centre.

3.1. Sequence-to-sequence (Seq2seq) model

Neural models of generation decode the natural language output – usually a sequence of tokens – from a vector representation of the semantic frame. In end-to-end dialogue systems, the vector representation may be the output of the preceding component in the pipeline. In standalone NLG systems like ours, the structured semantic frame input is transformed into a vector by an encoder of some sort. In previous work the vector representation is either manually constructed [14], or produced by an LSTM encoder (yielding a traditional seq2seq setup)[28]. In all experiments, we use a seq2seq encoder-decoder model with attention, first introduced in machine translation [29].

The input to the seq2seq model is a sequence of tokens $x_t, t \in \{0, \ldots, n\}$ that represent the dialogue act and associated arguments. Each x_t is associated with a learned embedding vector w_t of some fixed length. The encoder reads all the input vectors and encodes the sequence into a vector h_n. Specifically, at each time step t, we compute the hidden layer h_t from the input w_t and hidden vector at the previous time step h_{t-1}, following

$$h_t = f(x_t, h_{t-1})$$

As detailed in the next section, we formulate the input tokens to the encoder in multiple ways, for example, the slot values could be represented as independent tokens or jointly with the slot tags, or the input could be a flat frame or a plan that can include multiple sentences. The second phase of the sequence-to-sequence model, the decoder, is common to all experiments. The decoder LSTM generates each word of the sentence as conditioned on the context vector c_t. The probability distribution p_t for each candidate word at time t is calculated following

$$s_t = f(y_{t-1}, s_{t-1}, c_t)$$

$$p_t = softmax(s_t, y_{t-1})$$

where s_t is the hidden state of the decoder LSTM at time t. The context vector is a weighted average of the hidden states of the encoder:

$$c_t = \sum_{k=1}^{n} \alpha_{t,k} h_k$$

$$\alpha_{t,k} = exp(e_{t,k}) / \sum_{j=1}^{n} exp(e_{t,j})$$

$$e_{t,j} = v(s_{t-1}, h_t)$$

where v is usually implemented as a feed-forward network.

3.2. Extensions

The input to the seq2seq model is a sequential representation of a semantic frame, and the output is a sequence of tokens. We consider variations that demonstrate the influence of slot values on the surface form. In each setup, we select a strategy for

Table 5: *Input tokens for the various mention representations. In* SEQUENTIAL *and* JOINT*, there is a single vector w_i for each x_i, which is read by the encoder. In* CONCAT*, each $x_{i,j}$ has a unique $w_{i,j}$, and w_i is the concatenation of $w_{i,1}$ and $w_{i,2}$.*

Mention rep.	Input sequence					
SEQ	x_i	x_{i+1}	x_{i+2}	x_{i+3}	x_{i+4}	...
	decor	*decent*	service	good	cuisine	...
JOINT	x_i		x_{i+1}		x_{i+2}	
	⟨ decor, decent ⟩		⟨ service, good ⟩		⟨ cuisine, null ⟩	
CONCAT	$x_{i,1}$	$x_{i,2}$	$x_{i+1,1}$	$x_{i+1,2}$	$x_{i+2,1}$	$x_{i+2,2}$
	decor	decent	service	good	cuisine	null

mention representation (Table 5) and a strategy for plan supervision (Table 6). The mention representation strategy determines how a mention – a slot, value pair occurring in the semantic frame – is represented in the input sequence. The plan supervision strategy determines the amount of planning information, gleaned from the overall statistics of the human-generated examples, to be provided to the model at training and inference time.

3.3. Mention representation

A seq2seq model operates on a sequence of embeddings representing the input. All input sequences in our experiments have units denoting the dialogue act as a prefix and suffix, while intervening units represent slot-value pairs, or mentions. When mentions are delexicalizable, it is not necessary to supply the value to the model. In contrast, to generate utterances from this dataset, we expect that it will be important for the LSTM to be aware of the slot-values in the frame. We experiment with sequences using different units as different ways of associating a slot value with a slot name, in which the coupling of slots and values becomes increasingly explicit. These methods are summarized in Table 5. We use the following three representations:

- SEQUENTIAL: In a sequential representation, we produce a sequence in which a unit representing a slot is followed in the sequence by a unit representing its value, and the dependence between slots and values is encoded implicitly in the ordering.
- JOINT: In a joint representation, each unit in the sequence represents a combination of slot and value.
- CONCAT: A concatenated representation provides an intermediate variation, in which each slot and value corresponds to a unique embedding, but the slots and values are associated through concatenation of the embeddings.

We first performed experiments to determine the best mention representation, then, using this representation, continued to experiment with various forms of plan supervision.

3.4. Plan supervision

Traditionally, sentence planning occurs as a preliminary step, in which all the content of the utterance is considered at once, and decisions about ordering and grouping of slots are made at a global level. Subsequently, each individual sentence is generated with a subset of the required information [24]. We use a lightweight definition of sentence planning, in which the grouping of slot mentions into sentences, and the ordering of these sentences in the utterance together determine a plan. This requires no human annotation beyond tokenization and identifying slot mentions. We experiment with generating the entire sentence at once, as well as generating sub-parts of the plan individually. Table 6 provides examples of the following plan-supervision strategies:

Table 6: *Input tokens for the various plan supervision strategies. All slots are mentioned for* NONE*; slots are grouped into subsequences for* FLAT *and* FINE, POSITIONAL *adds tokens to indicate if a sentence is in the beginning or inside of an utterance.*

Plan sup.	Input tokens					
NONE	decor	decent	service	decent	quality	good
FLAT	decor	decent	service	decent		
	quality	good				
POSITIONAL	<B>	decor	decent	service	decent	
	<I>	quality	good			

- NONE: No plan is provided to the model, and the entire utterance is generated from a representation of the entire frame.
- FLAT: The slot groupings are derived externally using corpus statistics, and each sentence in the utterance is produced using a separate input sequence, with **no** information on the context.
- POSITIONAL: We add a token to the input sequence both at training and inference time to indicate the sentence's position in the utterance, (*beginning* or *inside*).

4. Data

Our experiments use utterances in the restaurant domain, using a schema similar to that of Zagat ratings [24]. The slots are shown in Table 3 . Slots are categorized according to the values that they can take. Three slots have categorical values, such as restaurant names or city names. The set of values is not necessarily closed, and there is no particular relation between the values in the set. Four slots take scalar values. Each utterance recommends a restaurant, mentioning some of its qualities. We deliberately curtail the number of slots and possible values, as our intention is to isolate the problems of slot aggregation and discourse structuring. Although the dataset is simple in this respect, it is rich in variations of slot combinations.

4.1. Dataset schema

An NLG system must generally be able to produce multiple dialogue acts, e.g. `request`, `inform`, and others. Here we focus on `recommend` dialogue acts with multiple slot mentions, where the issues of aggregation and discourse structure are most pertinent, e.g. as in the examples in Tables 1 and 4.

We construct semantic frames so that every frame mentions all three categorical slots and three scalar slots, totaling **6 slots per frame**. See Table 3. Here, each scalar value slot (*food quality*, *decor*, *service* and *value*) takes a value from *mediocre*, *decent*, *good*, or *excellent*. We cover the entire space of slot combinations and value assignments, resulting in $4 \cdot 4^3 = 256$ unique frames. As a result, the dataset contains examples where all slot values are different, and ones where two or more values are the same. Previous work also experiments with utterances mentioning up to 8 slots. [14, 30]. However since slots only took categorical values with disjoint ranges, aggregation across slot values could not occur.

Thus, our dataset is novel in three ways: (1) all utterances contain 6 slots, which is relatively more challenging for an LSTM as compared to 1-2 slots used in some previous work[14], (2) slots share the same value space, which supports aggregation; (3) the emphasis on scalar valued slots to elicit the use of discourse cues.

4.2. Data collection

We elicited 7 surface realizations for each semantic frame using Mechanical Turk. This resulted in many different highly natural

Table 7: *Results for mention representation and plan supervision. * indicates* CONCAT *values significantly better than* SEQ*; † indicates significantly better than* HUMAN-S *(both $p < 0.05$). Bold values of* POSITIONAL *are significantly better than* NONE*. ($p < 0.05$).*

		Slot Prec.	Scalar Prec.	Slot Rec.	Naturalness	Syntax	Overall	# uniq. sents
Mention representation	SEQ	1.0	0.6	92.24%	2.72	3.04	2.75	100
	JOINT	1.0	0.86	86.26%	2.42	2.75	2.41	101
	CONCAT	1.0	0.98	**95.33%***	**2.80†**	2.98	**2.87***	126
	HUMAN-G	1.0	1.0	94.98%	2.65	2.93	2.76	313
	HUMAN-S	1.0	1.0	95.29%	2.64	2.87	2.79	325
Plan supervision	NONE	-	-	97.22%	2.69	2.80	2.68	126
	FLAT	-	-	96.99%	2.68	2.93	2.74	152
	POSITIONAL	-	-	96.99%	2.74	**3.02**	**2.81**	144
	HUMAN-S	-	-	96.01%	2.72	2.91	2.83	325

utterances, where Turkers produced many different phrasings such as those in Rows 1, 2, 3, and 4 in Table 1. Rows 8 and 9 in Table 4 shows how Turkers both naturally aggregated slots, without being told to explicitly, and used discourse cues such as *all*, *both* and *but*. They also appropriately used anaphoric forms such as *they* and *it* as shown in Rows 8 and 9. After filtering, the dataset contains 1662 utterances, divided into train, development and test in the ratio 8:1:1.

5. Experiments

5.1. Evaluation

Our evaluation experiments solicit judgements from raters and trained annotators for both objective and subjective metrics. We attempt to have more detailed measures than what automatic metrics such as BLEU provide[31, 32, 33]. Turkers were presented with HITs containing three utterances for a particular sentence plan and asked for judgements on each utterance individually, and then asked to compare the three utterances overall.

Subjective evaluation. Turkers rated each utterance on a 5 point Likert scale as to whether it was *natural* and *grammatically correct*. We then asked them to comparatively evaluate three utterances at a time for their *overall goodness*, again on a 5 point Likert scale. Ratings ranged from 0 (worst) to 4 (best).

Objective evaluation. Turkers were asked to indicate if each slot occurred with the correct value; this is reported in Table 7 as *slot recall*. We also asked them separately to report whether there were any extraneous slots present (*slot precision*). As this was difficult for the raters to report accurately, one of the authors (a native speaker) judged *slot precision* for a sample of 50 utterances produced by each model. We also separately evaluated the correctness of scalar values assignment to slots in this sample. These are reported as *Scalar precision* in Table 7.

5.2. Results and Analysis

We first compared various mention representation strategies in an LSTM setting akin to NONE plan supervision. Along with the generated utterances, the raters were presented with human-generated alternatives of two kinds: HUMAN-GENERAL and HUMAN-SPECIFIC. In each of these, given a frame to generate from, we attempt to select an utterance from the training set that matches this frame. In HUMAN-SPECIFIC, we attempt to match both slots and values, whereas in HUMAN-GENERAL we simply attempt to match slots. This is essentially template-based generation, where a template is induced from the human utterances, and used for realization, as in other work [34]. Table 7 summarizes the results of our experiments.

In this evaluation, we found that CONCAT outperformed the simple sequential representation significantly in terms of recall and overall rating. In addition, CONCAT surprisingly outperformed HUMAN-S significantly in naturalness. Thus, we used CONCAT as the mention representation strategy in the plan supervision experiments.

In the plan supervision experiments, we compared the two forms of plan supervision with the baseline of NONE, which was equivalent to CONCAT in previous experiments, and the human-generated set HUMAN-S. We found that the POSITIONAL model was able to improve significantly on syntax ratings, while maintaining overall and naturalness scores.

Comparison to human-generated sentences. We note that in some instances we appear to outperform humans, for example, on naturalness (CONCAT). The high naturalness score of CONCAT can be attributed to the model gravitating towards 'safe' – commonly occurring, grammatical – utterance patterns. In contrast, the data for HUMAN-S is generated by crowd workers. Thus, it has more stylistic variation, and may include lower quality examples (such as if a rater forgot to mention a slot). This results in higher variance in the scores for this baseline. Our models have a smoothing effect, which, while reducing stylistic variation, also allows them to produce high-quality utterances when trained on a dataset of inconsistent quality.

Scalar Precision. Our setup is novel in that scalar values are shared between slots. The *scalar precision* measure evaluates whether the correct values are associated with each slot for the scalar slots. Since this is not a subjective judgement, we had a trained annotator inspect a random sample of 50 sentence plans and their model outputs to gauge scalar precision. *None* of the realizations contained additional slots with values different than those specified in the semantic frame. However, CONCAT was significantly less likely than the SEQ model to have scrambled slot values across slots, as shown in Table 7.

Diversity. In experiments for mention representation, we find that the model converges to use sentence plans that appear frequently in the training data. While this results in high scores on syntax, the diversity of utterances produced by this model is lacking. In particular, CONCAT and HUMAN-S produced a total of 126 and 325 unique sentences in the test set, respectively. We also note that of the 1662 utterances in the dataset, the sentence plans of 524 utterances are accounted for by the 4 most frequent sentence plans. This is addressed in our experiments with plan supervision. We found that FLAT improved the number of distinct sentences produced by 20%.

6. Conclusions

In this work, we propose slot-value informed sequence-to-sequence models with attention for language generation in dialogue systems, where the slot values in the system action, as well as the slot tags are part of the input sequence. We investigate three ways of representing the values in the input sequence. Furthermore, we study if the models can learn to plan the sentences given the limited amount of training data, or if they would benefit from being presented a plan in the form of multiple sentences as part of the input. In subjective evaluations, we show that the slot values represented as vectors concatenated to vectors of slot names results in best overall quality. Furthermore, adding a plan to the input further improves quality and diversity.

7. References

[1] P. Shah, D. Hakkani-Tür, and L. Heck, "Interactive reinforcement learning for task-oriented dialogue management," in *NIPS 2016 Deep Learning for Action and Interaction Workshop*, 2016.

[2] M. A. Walker, "An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email," *Journal of Artificial Intelligence Research*, vol. 12, pp. 387–416, 2000.

[3] L. Heck and D. Hakkani-Tür, "Exploiting the semantic web for unsupervised spoken language understanding," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 228–233.

[4] R. Lowe, N. Pow, I. Serban, and J. Pineau, "The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems," in *Proc. of the SIGDIAL 2015 Conference: The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015.

[5] R. Abbott, B. Ecker, P. Anand, and M. Walker, "Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it," in *Language Resources and Evaluation Conference, LREC2016*, 2016.

[6] R. Higashinaka, N. Kobayashi, T. Hirano, C. Miyazaki, T. Meguro, T. Makino, and Y. Matsuo, "Syntactic filtering and content-based retrieval of twitter sentences for the generation of system utterances in dialogue systems," in *Situated Dialog in Speech-Based Human-Computer Interaction*. Springer, 2016, pp. 15–26.

[7] H. Sugiyama, T. Meguro, R. Higashinaka, and Y. Minami, "Open-domain utterance generation for conversational dialogue systems using web-scale dependency structures," in *Proc. SIGDIAL*, 2013, pp. 334–338.

[8] D. Hakkani-Tür, G. Tur, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-domain joint semantic frame parsing using bi-directional rnn-lstm," in *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*, 2016.

[9] N. Dethlefs, H. Cuayáhuitl, H. Hastie, V. Rieser, and O. Lemon, "Cluster-based prediction of user ratings for stylistic surface realisation," *EACL 2014*, p. 702, 2014.

[10] V. Rieser and O. Lemon, *Reinforcement learning for adaptive dialogue systems: a data-driven methodology for dialogue management and natural language generation*. Springer Science & Business Media, 2011.

[11] F. Mairesse and M. A. Walker, "Controlling user perceptions of linguistic style: Trainable generation of personality traits," *Computational Linguistics*, 2011.

[12] R. Higashinaka, M. A. Walker, and R. Prasad, "An unsupervised method for learning generation lexicons for spoken dialogue systems by mining user reviews," *ACM Transactions on Speech and Language Processing*, vol. 4, no. 4, 2007.

[13] F. Charras, G. Dubuisson Duplessis, V. Letard, A.-L. Ligozat, and S. Rosset, "Comparing system-response retrieval models for open-domain and casual conversational agent," in *Workshop on Chatbots and Conversational Agent Technologies*, 2016.

[14] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, "Semantically conditioned lstm-based natural language generation for spoken dialogue systems," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[15] R. Lebret, D. Grangier, and M. Auli, "Generating text from structured data with application to the biography domain," in *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[16] M. Walker and O. Rambow, Eds., *Computer Speech and Language: Special Issue on Spoken Language Generation*. Academic Press, 2002, vol. 16: 3-4.

[17] S. M. Lukin, L. I. Reed, and M. Walker, "Generating sentence planning variations for story telling," in *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015, p. 188.

[18] A. Ritter, C. Cherry, and W. B. Dolan, "Data-driven response generation in social media," in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2011, pp. 583–593.

[19] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan, "A neural network approach to context-sensitive generation of conversational responses," in *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2015.

[20] O. Dušek and F. Jurcıcek, "Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings," in *The 54th Annual Meeting of the Association for Computational Linguistics*, 2016, p. 45.

[21] R. Barzilay and M. Lapata, "Aggregation via set partitioning for natural language generation," in *Proceedings of NAACL 2006: Conference of the North American Chapter of the Association of Computational Linguistics*, 2006, pp. 359–366.

[22] M. Walker, O. Rambow, and M. Rogati, "Training a sentence planner for spoken dialogue using boosting," *Computer Speech and Language: Special Issue on Spoken Language Generation*, vol. 16, no. 3-4, pp. 409–433, 2002.

[23] R. Barzilay and L. Lee, "Bootstrapping lexical choice via multiple-sequence alignment," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 2002, pp. 164–171.

[24] M. A. Walker, A. Stent, F. Mairesse, and R. Prasad, "Individual and domain adaptation in sentence planning for dialogue," *Journal of Artificial Intelligence Research (JAIR)*, vol. 30, pp. 413–456, 2007.

[25] A. Stent, "A conversation acts model for generating spoken dialogue contributions," *Computer Speech and Language: Special Issue on Spoken Language Generation*, 2002.

[26] I. Langkilde and K. Knight, "Generation that exploits corpus-based statistical knowledge," in *Proc. of COLING-ACL*, 1998.

[27] C. Paris and D. Scott, "Stylistic variation in multilingual instructions," in *The 7th International Conference on Natural Language Generation*, 1994.

[28] O. Dušek and F. Jurcicek, "A context-aware natural language generator for dialogue systems," in *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2016.

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[30] J. Novikova, O. Lemon, and V. Rieser, "Crowd-sourcing nlg data: Pictures elicit better data," in *International Conference on Natural Language Generation*, 2016.

[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "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, 2002, pp. 311–318.

[32] A. Belz and E. Reiter, "Comparing automatic and human evaluation of nlg systems." in *EACL*, 2006.

[33] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "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)*, 2016.

[34] G. Angeli, P. Liang, and D. Klein, "A simple domain-independent probabilistic approach to generation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 502–512.