End-to-End Trainable System for Enhancing Diversity in Natural Language Generation

Jan Deriu
Zurich University of Applied Sciences
jan.deriu@zhaw.ch

Mark Cieliebak
Zurich University of Applied Sciences
mark.cieliebak@zhaw.ch

Abstract

Natural Language Generation plays an important role in the domain of dialogue systems as it determines how the users perceive the system. Recently, deep-learning based systems have been proposed to tackle this task, as they generalize better and do not require large amounts of manual effort to implement them for new domains. However, deep learning systems usually produce monotonous sounding texts. In this work, we present our system for Natural Language Generation where we control the first word of the surface realization. We show that with this simple control mechanism it is possible to increase the lexical variability and the complexity of the generated texts. For this, we apply a character-based version of the Semantically Controlled Long Short-term Memory Network (SC-LSTM), and apply its specialized cell to control the first word generated by the system. To ensure that the surface manipulation does not produce semantically incoherent texts we apply a semantic control component, which we also use for reranking purposes. We show that our model is capable of generating texts that are more sophisticated while decreasing the number of semantic errors made during the generation.

1 Introduction

In this paper, we describe our end-to-end trainable neural network for producing natural language descriptions from meaning representations (MR). We focus on generating more diverse and interesting texts since the texts generated by state-of-the-art systems produce rather monotonous texts. Recently, data-driven natural language generation (NLG) systems have shown great promise, especially as they can be easily adapted to new data or domains. End-to-end systems based on deep learning can jointly learn sentence planning and sentence realization from unaligned data. However, deep-learning based approaches require large amounts of data to be trained efficiently which is not always readily available. For this reason, we train our system on the publicly available E2E dataset provided by (Novikova et al., 2017) for the *E2E NLG Challenge 2017*[1] which provides pairs of MRs and several human generated reference utterance for the restaurant domain. This dataset is the first to provide large amounts of training data with an open vocabulary and complex syntactic structures. These properties pose a further challenge for training the system as the large variety of formulations for an attribute-value pair does not allow to simply replacing the attribute with a token.

A recurrent problem, which we found with the existing solutions for NLG, are the rather monotonous texts they generate. Most generated sentences follow the same structure, i.e. they start with the restaurant name and they use the same formulation to express each attribute value.

In this work, we focus on how to exploit the open vocabulary and the complex syntactic structures to generate sentences that are more sophisticated. For this, we extend the Semantically Conditioned Long Short-term Memory Network (SC-LSTM) proposed by (Wen et al., 2015b) with surface features as well as an additional semantic control mechanism similar to (Hu et al., 2017). Furthermore, we train the SC-LSTM on character tokens instead of word tokens to avoid sampling over a potentially large vocabulary. Since the E2E data expresses a single attribute value pair with a larger variety of possible formulations, a simple delexicalization of the utterance is not possible. This fact also increases the difficulty of evaluating the utterances for their correctness. Thus, we intro-

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

duce a semantic reranking procedure based on the classifiers that are trained as part of the semantic control mechanism.
We report the evaluation results provided by the *E2E NLG Challenge 2017*, these includes automatic metrics as well as a human evaluation. The evaluation showed, that the automatic metrics rate the more sophisticated sentences lower than the standard sentences. In the human evaluation our approach ranked in the 2nd out of four clusters for *quailty* and in the 3rd out of five clusters for naturalness.

2 Task Definition

Natural language generation for dialogue systems describes the task of converting a meaning representation (MR) into an utterance in a natural language. In the context of a dialogue system, the dialogue manager returns the output in the form of structured data called meaning representations. For a more in depth treatment of dialogue systems refer to (Rieser and Lemon, 2011). Usually, they contain a dialogue act which defines the action (e.g. inform, recommend) and a list of slot (or attribute) value pairs which define the content of the utterance. The E2E training data consist of 50k instances in the restaurant domain (see Table 1 for an example), where one instance is a pair of a MR and an example utterance or reference.

MR	name[Alimentum], food[Chinese], priceRange[20-25], area[riverside], familyFriendly[yes]
REF1	You can find average-priced Chinese food by the river at the kid-friendly Alimentum.
REF2	Alimentum is a Chinese restaurant located in the riverside area. Meals are in the 20-25 pound range and it is kid friendly.
REF3	Alimentum provides Chinese food in the 20-25 price range. It is located in the riverside.

Table 1: Example of one meaning representation (MR) with three corresponding references (REF) from the training data.

The data is split into training, development and test in a 76.5%-8.5%-15%-ratio. Each MR consists of 3-8 attributes and their values, see Table 2 for the domain ontology. The split ensures that the MRs in the different dataset-splits are distinct. The dataset contains an open vocabulary and more complex syntactic structures than other similar datasets, as shown in the dataset definition (Novikova et al., 2017).

Attribute	Type	Example Values
name	verbatim string	Alimentum, ..
eatType	dictionary	restaurant, pub, coffee shop
familyFriendly	boolean	yes, no
food	dictionary	Italian, French, English, ...
near	verbatim string	Burger King
area	dictionary	riverside, city center
customerRating	dictionary	1 of 5, 3 of 5, 5 of 5, low, average, high
priceRange	dictionary	<£20, £20-25, >£30 cheap, moderate, high

Table 2: Domain ontology of the E2E dataset.

3 Model

The goal of our model is to generate a text while providing the ability of controlling various semantic and syntactic properties of this text. Our model has two components: i) the generator and ii) multiple semantic control classifiers. An overview of the model is given in Figure 1.
We use the Semantically Conditioned Long Short-term Momory Network (SC-LSTM) proposed by (Wen et al., 2015b) as our generator which has a specialized cell to process the one-hot encoded MR-vector. A semantic control classifier for an attribute is a classifier trained to detect which of its possible values is rendered in the text. We pretrain the classifiers on the provided labelled data; we provide an extra label that represents the presence or absence of the attribute. During the training phase of the generator, we fix the classifier parameters, and their loss signal is back-propagated through the generator alongside the reconstruction loss. During the testing phase, we use the classifiers to rerank the generated sentences.

Semantically Conditioned LSTM The SC-LSTM (Wen et al., 2015b) extends the original LSTM (Hochreiter and Schmidhuber, 1997) cell with a specialized cell which processes the MR. The MR is represented as a one-hot encoded MR-vector d_0 which represents the value for each attribute. This cell assumes the task of the sentence planner, as it treats the MR-vector as a checklist to ensure the information is fully represented in the utterance. The cell acts as a forget gate keeping track of which information has already been consumed. Let $w_t \in \mathbb{R}^M$ be the input vector at time t, $d_t \in \mathbb{R}^D$ the MR-vector at time t, and N be the number of units of an SC-LSTM cell, then the

formulation of the forward pass is defined as:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ r_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{W}_{5n,2n} \begin{pmatrix} w_t \\ h_{t-1} \end{pmatrix}$$

$$d_t = r_t * d_{t-1}$$
$$c_t = i_t * g_t + f_t * c_{t-1} + \tanh(W_d d_t)$$
$$h_t = o_t * \tanh(c_t)$$

where σ is the sigmoid function, and $i_t, f_t, o_t, r_t \in [0,1]^N$ are the input, forget, output, and MR-reading gates. The weights $\mathbf{W}_{5n,2n}$, and $W_d \in \mathbb{R}^{D \times M}$ are the model parameters to be learned. The prediction of the next token is performed by sampling from the probability distribution:

$$w_t \sim P(w_t | w_{0:t-1}, d_t) = \text{softmax}(W_s h_t)$$

where $W_s \in \mathbb{R}^{N \times M}$ is a weight matrix to be learned during training. During the training procedure the inputs to the SC-LSTM are the original tokens w_t from the training set. On the other hand, when generating new utterances we use the previously generated token as input to generate the next token.

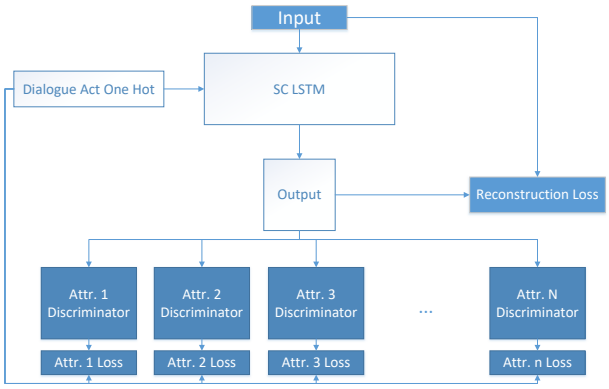

Figure 1: Overview of the system. The MR-vector along with the correct reference sentence are used as input to the SC-LSTM during training. The output of the SC-LSTM is fed into the semantic control classifiers, and their loss is used during back-propagation alongside the reconstruction loss.

Loss To ensure that the SC-LSTM consumes the MR correctly two conditions are defined: i) the MR-vector at the last time step d_T has to be zero, which ensures that all the required information has been rendered, and ii) the gate should not consume too much of the dialogue act in one time step, i.e. the difference $\|d_t - d_{t-1}\|$ should be minimised. From these criteria, the reconstruction loss is adapted to:

$$F(\theta) = \sum_t p_t^T log(y_t) + \|d_T\| + \sum_{t=0}^{T-1} \eta \xi^{\|d_t - d_{t-1}\|}$$

where the first term is the reconstruction error which sums the cross-entropy loss for each time step and the following two terms ensure the two criteria defined above.

Semantic Control For each attribute a we train a CNN-based classifier D_a that classifies which of the possible values for the attribute a is rendered in the utterance or if the attribute is present in the utterance at all. We pretrain the classifier on the training set. During the training of the generator, we fix the classifier weights and pass the output of the generator into the classifier, which return the loss. Let $F(\theta_a)$ be the categorical cross entropy loss of the classifier for attribute a. We further adapt the reconstruction loss for the generator to:

$$F(\theta) = \sum_t p_t^T log(y_t) + \|d_T\| + \sum_{t=0}^{T-1} \eta \xi^{\|d_t - d_{t-1}\|} + \sum_{a \in A} F(\theta_a)$$

Thus, the generator receives feedback on the semantic correctness of its output.

4 First Word Control

We observe that the vanilla SC-LSTM produces utterances that express the same syntactic structure, especially all the generated utterances start with the "X-name" token. There are two main reasons for this behaviour. First, from the formulation of the sequence inference it is obvious that the prediction of the next token is dependent on the previously produced tokens. Thus, the first word used in an utterance determines how the rest of the utterance is produced. The second reason is that in the training set 59% of the utterances start with the "X-name" token and only 7% start with the word "There" (see Figure 2).

Without additional information, the model optimizes to generate the utterance which yields the lowest average loss, which are the most common utterances in the training set. To generate more uncommon utterances, we provide the model with

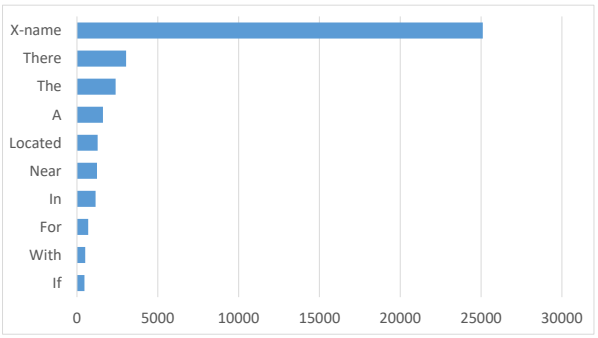

Figure 2: Number of occurrences of the ten most frequent first words in the training set.

the information about the first word in the utterance. For this, we select all the words that appear more than $t = 60$ times as first word, which results in a list of $n = 26$ different words. We then extend the MR-vector by adding a one-hot encoded vector $f_0 \in \mathbb{R}^{n+1}$:

$$f_0 = \begin{cases} f_0[\text{first word index}] = 1, & \text{first word in list} \\ f_0[n+1] = 1, & \text{first word not in list} \end{cases}$$

5 Experimental Setting

The goal for our application is to generate descriptions for restaurants. The dataset contains 50k utterances for 5,751 different MRs. On average, each MR is composed of 5.43 attributes and there are 8.1 different references for each MR on average. Refer to Table 3 for the data-split. For the evaluation, we report various corpus-based metrics: BLEU-4 (Papineni et al., 2002), which computes the precision of the n-grams in the generated candidate with multiple reference utterances; NIST (Doddington, 2002) which extends the BLEU score by taking into account the informativeness of an n-gram; METEOR (Lavie and Agarwal, 2007), which is based on the harmonic mean of unigram precision and recall and takes into account synonyms; ROUGE-L (Lin, 2004), which compares the longest co-occurring subsequence; and CIDEr (Vedantam et al., 2015), which is based on TF-IDF scoring on n-grams.
To assess the complexity of the generated utterances, we employ the Lexical Complexity Analyser (Lu, 2012).

	Training	Validation	Testing
REF	42,067	4,672	4,693
MR	4574	547	630

Table 3: Data split by number of references and number of MRs.

Preprocessing Since we work on character level, we treat each utterance as a string of character, where each character is represented as a one-hot encoded vector. Since the training set expresses a high diversity in the formulations of some attribute value pairs, we can only replace the *name* and *near* values with the tokens 'X-name" and "X-near" respectively. To generate the first-word features, we apply the Spacy-API[2] for tokenization.

System Setup We train the SC-LSTM and the classifiers using AdaDelta (Zeiler, 2012) to optimize the loss function. The classifiers are pre-trained on the training set using early stopping to avoid over-fitting. To train the SC-LSTM, we fix the weights of the classifiers and feed the output of the SC-LSTM into the classifiers. Since the classifiers are trained on one-hot encoded character representations but the SC-LSTM returns softmax-probabilities, we apply a softmax with decreasing temperature as proposed in (Hu et al., 2017) to approximate the discrete representation. For the LSTM cell we use a hidden state of size 1024 and apply dropout as suggested in (Yarin and Ghahramani, 2016). For the classifier we use a 2-layer CNN with 256 kernels of length 3.
We use our character-based version of the SC-LSTM by (Wen et al., 2015b) as baseline (*Base Model*) and the models where we control the first word (*primary 1 GAN, primary 2 Vanilla*). To assess the impact of the semantic control classifiers we compare a model trained by back-propagating the classifier-losses *primary 1 GAN* and a model trained without back-propagating the classifier losses *primary 2 Vanilla*.

Output Selection Through the syntactic manipulations of the utterances there are cases where the manipulation contradicts the MR. For example, if the MR has no information about the area or near restaurants, beginning the sentence with "Located" or "Near" would result in the generation of redundant information. To ensure that the final utterance for a given MR is correct, we produce one output for each of the 26 possible first words and select those, which received the highest correctness score from the classifiers. From this set of "most correct" utterances, we sample uniformly at random the final output utterance.

[2]https://spacy.io/

System Name	BLEU	NIST	METEOR	ROUGE-L	CIDEr
Base Model	**0.65**	**8.34**	**0.44**	**0.67**	**2.14**
primary 1 GAN	0.58	8.02	0.43	0.59	1.81
primary 2 Vanilla	0.60	8.13	0.43	0.61	1.91

Table 4: Results of the automatic metrics for the five tested systems.

System Name	BLEU	NIST	METEOR	ROUGE-L	CIDEr
Base Model	**0.69**	**8.31**	**0.47**	**0.71**	**2.22**
primary 1 GAN	0.59	7.85	0.44	0.61	1.84
primary 2 Vanilla	0.60	7.99	0.45	0.62	1.93

Table 5: Validation results of the automatic metrics for the five tested systems.

System Name	ERR	Missing	Redundant
Base Model	7.3%	5.3%	2.0%
primary 1 GAN	5.0%	**3.8%**	0.12%
primary 2 Vanilla	**4.6%**	4.5%	**0.01%**

Table 6: Error analysis based on the ERR score, the number of missing and redundant attribute values.

name	eatType	price	rating	near	food	area	fam.
1.0	0.97	0.90	0.84	0.99	0.95	0.94	0.91

Table 7: Validation Accuracy scores for each classifier.

6 Results

In this section we present the results of the evaluation. We apply the metrics used by the *E2E-NLG* challenge. Additionally we evaluate the lexical complexity to assess the impact of our proposed control mechanism.

6.1 Evaluation Metrics

We report the scores for the automatic evaluation. This includes the metrics BLEU, ROUGE-L, METEOR, NIST, and CIDEr score. Table 4 and Table 5 show that the surface manipulation leads to a decrease in all of these scores. The best score is achieved by the *Base Model*. The models achieve significantly lower scores when we manipulate the first word: in both cases the BLEU score drops by $5-7$ points and other scores accordingly. Only the METEOR score, which takes semantic similarity among words into account, remains stable.
The reason for this drop is that in 59% of the training references the first word is the "X-name" token, thus, generating an utterance that starts with the "X-name" token yields a higher token overlap with the reference. On the other hand, the utterances where we controlled the first word make less use of the "X-name" token for the first word. In fact, the "X-name" token was only used in 3% of the cases as the first word. Thus, there is a lower probability that the generated utterance overlaps with one of the references, hence, the evaluation metrics yield lower results.

6.2 Classifier Performance

Since we use the classifiers to evaluate the generated sentences, it is important to assess the quality of these classifiers. Table 7 shows the validation accuracy score for each of the classifiers. We note that all classifiers have a score greater than 0.9 except for the *customer rating*. The errors of the *customer rating* and the *price* classifiers stem from the semantic equivalence between the numerical and the verbal values which were used interchangeably in the references.

6.3 Correctness

We evaluate the semantic correctness using the ERR $= \frac{p+q}{N}$ score proposed by (Wen et al., 2015a), where p is the number of redundant or incorrect values, q the number of missing values and N the number of attributes in the MR. We report the scores for the missing and redundant values separately. Table 6 shows that the use of First Word Control reduces the error rate, and in particular, the rate of missing values is greatly reduced. The reason is that with manipulation of the first word, the model has more possibilities for making the utterance. Thus, the reranking has the possibility to select from multiple utterances. We also note that the SC-LSTM with semantic control classifier extensions generates significantly more redundant information than without the extensions. The largest part of the errors arise when the *eatType* attribute is not specified, but the model renders this information regardless, see Table 10.

6.4 Lexical Complexity Analysis

To assess the lexical richness we use the Lexical Complexity Analyser (Lu, 2012), which is a collection of metrics to measure various aspects of lexical richness. Table 8 displays the various scores for the different systems and metrics.
We observe that the number of different words (NDW) increases when we control for the first word. We also note that the *primary 1 GAN* has a higher NDW than the *Base Model*. In fact, when controlling the first word we almost double the NDW w.r.t the *Base Model*.
The lexical sophistication (LS) computes the proportion of word types that do not appear in the 2000 most frequent words generated from the

System Name	NDW	LS	CVS	TTR	MSTTR	LV	VV	NV
Base Model	98	0.11	0.02	$5e^{-3}$	0.57	$5e^{-3}$	$6e^{-3}$	$5e^{-3}$
primary 1 GAN	162	0.09	0.08	$9e^{-3}$	0.61	$12e^{-3}$	$28e^{-3}$	$9e^{-3}$
primary 2 Vanilla	135	0.08	0.04	$8e^{-3}$	0.61	$8e^{-3}$	$18e^{-3}$	$7e^{-3}$

Table 8: Results of the lexical complexity analysis.

British National Corpus. We observe that all the models have a ratio of about 10%, with the uncontrolled models having the highest percentages. However, when reporting on the verb sophistication (CVS, verbs not in the list of the 200 most frequent verbs), we observe that the controlled models perform slightly better than the *Base Model*.
To explore the lexical variation we use various metrics: the type-token ratio (TTR) measures the ratio between number of tokens and number of words in a text. It is sensitive to the size of the text, thus, we also report the mean segmental TTR (MSTTR), which divides the text into successive segments and computes the average TTR of these segments. The results show that for both TTR and MSTTR the controlled models have higher scores with values of 0.61 each. We also report the lexical word-variation (LV), which computes the ratio between number of different lexical tokens and total number of lexical tokens (lexical words: nouns, adjectives, verbs and adverbs). We observe again that the controlled models display a higher LV. The same pattern is notable for the verb-variation score and the noun variation score, which are the ratio of number of different verbs to total number of verbs and nouns, respectively.

6.5 Human Evaluation

We submitted both the *primary 1 GAN* and the *primary 2 Vanilla* systems to the *E2E NLG Challenge 2017* challenge, where they were evaluated by humans. The human subjects evaluated the system outputs for *quality* (grammatical correctness, fluency, adequacy, etc.) and *naturalness* (extent to which the utterance could have been produced by a native speaker). Both our submitted systems rank 2^{nd} out of four clusters for *quailty* and in the 3^{rd} out of five clusters for naturalness. For a complete analysis of the evaluation procedure consult (Dušek et al., 2018).

6.6 Qualitative Evaluation

To highlight the potential and the limitations of our approach, we look at some representative examples. In Table 9, we compare the outputs of the *Base Model* and the First Word Control. We observe that the structure of the utterances generated by the *Base Model* remains the same as we adapt one attribute. For instance, the utterance generated with the *rating-attribute* set to *1/5* is the same as when the attribute is set to *3/5*. On the other hand, through the manipulation of the first word the output becomes more diverse.

However, controlling the first word can lead to conflicting situations where the MR and the first word contradict each other. We see in Table 10 two examples where no information about the *area* or *near* was asked, but since we controlled the first word to be "Located" or "Near", the model rendered redundant information. We avoid this problem by generating an utterance for each first word and use the reranking to choose the most correct utterance.

7 Conclusion

In this work, we presented an end-to-end trainable deep-learning based system for natural language generation. We showed that it is possible to generate texts that are more sophisticated with a simple control mechanism. The evaluation revealed that the measured lexical diversity, the syntactic complexity as well as the semantic correctness significantly increased when manipulating the first word of an utterance. Furthermore, the evaluation showed that the standard metrics for evaluating an NLG system are not able to capture these manipulations. We observed a decrease in these metrics when controlling the surface realization, even though the utterances are semantically equivalent.

name[The Punter], area[riverside] familyFriendly[yes] rating[1/5]	V	The Punter is a kid friendly restaurant in the riverside area with a customer rating of 1 out of 5.
	F	A kid friendly restaurant called The Punter is located in the riverside area and has a customer rating of 1 out of 5.
name[The Punter], area[riverside] familyFriendly[yes] rating[3/5]	V	The Punter is a kid friendly restaurant in the riverside area with a customer rating of 3 out of 5.
	F	With a customer rating of 3 out of 5, The Punter is a kid friendly restaurant located in the riverside area.
name[Alimentum], area[riverside] familyFriendly[yes] near[N/A]	V	Alimentum is a family-friendly restaurant in the riverside area.
	F	If you are looking for a family friendly place there is a restaurant called Alimentum in the riverside area.
name[Alimentum], area[riverside] familyFriendly[yes] near[Burger King]	V	Alimentum is a family-friendly restaurant in the riverside area near Burger King.
	F	A family friendly place is Alimentum. It is located near Burger King in the riverside area.

Table 9: Sample output of the vanilla SC-LSTM (V) and the First Word Control (F) for four different MRs where one attribute-value is changed.

name[The Wrestlers] rating[1/5] familyFriendly[yes]	Children friendly The Wrestlers rates 1 out of 5.
	Near the riverside is a kid friendly place called The Wrestlers with a rating of 1 out of 5.
	Located near the river, The Wrestlers is a kid friendly restaurant with a customer rating of 1 out of 5.

Table 10: Utterances generated for the same MRs by controlling the first word highlighting some of the problems that arise when the MR and the first word are contradicting each other. The red colour denotes redundant information.

References

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, HLT '02, pages 138–145.

Ondrej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *(in prep.)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* pages 1735–1780.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of tex. *International Conference on Machine Learning* pages 1587–1596.

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.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. Association for Computational Linguistics, Barcelona, Spain, pages 74–81.

Xiaofei Lu. 2012. The relationship of lexical richness to the quality of esl learners oral narratives. *The Modern Language Journal* 96(2):190–208.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Saarbrücken, Germany. ArXiv:1706.09254.

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.

Verena Rieser and Oliver Lemon. 2011. *Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation*. Springer Publishing Company, Incorporated.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015a. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. Association for Computational Linguistics, pages 275–284.

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 *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Gal Yarin and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. *Advances in neural information processing systems* pages 1019–1027.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .