

Unsupervised Natural Language Generation with Denoising Autoencoders

Markus Freitag and Scott Roy

Google AI

{freitag,hsr}@google.com

Abstract

Generating text from structured data is important for various tasks such as question answering and dialog systems. We show that in at least one domain, without any supervision and only based on unlabeled text, we are able to build a Natural Language Generation (NLG) system with higher performance than supervised approaches. In our approach, we interpret the structured data as a corrupt representation of the desired output and use a denoising auto-encoder to reconstruct the sentence. We show how to introduce noise into training examples that do not contain structured data, and that the resulting denoising auto-encoder generalizes to generate correct sentences when given structured data.

1 Introduction

Natural Language Generation (NLG) is the task of generating text from structured data. Recent success in Deep Learning motivated researchers to use neural networks instead of human designed rules and templates to generate meaningful sentences from structured information. However, these supervised models work well only when provided either massive amounts of labeled data or when restricted to a limited domain. Unfortunately, labeled examples are costly to obtain and are non-existent for many domains. Conversely, large amount of unlabeled data are often freely available in many languages.

A labeled example is given in Table 1. One labeled example consists of a set of slot pairs and at least one golden target sequence. Each slot pair has a slot name (e.g. "name") and a slot value (e.g. "Loch Fyne"). In this work, we present an unsupervised NLG approach that learns its parameters without the slot pairs on target sequences only. We use the approach of a denoising auto-encoder (DAE) (Vincent et al., 2008) to train our

name	type	food	family friendly
Loch Fyne	restaurant	Indian	yes

Table 1: Three possible correct target sequences for the structured data above: (a) There is an Indian restaurant that is kids friendly. It is Loch Fyne. (b) Loch Fyne is a well-received restaurant with a wide range of delicious Indian food. It also delivers a fantastic service to young children. (c) Loch Fyne is a family friendly restaurant providing Indian food.

model. During training, we use corrupt versions of each target sequence as input and learn to reconstruct the correct sequence using a sequence-to-sequence network (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015). We show how to introduce noise into the training data in such a way that the resulting DAE is capable of generating sentences out of a set of slot pairs. Taking advantage of using unlabeled data only, we also incorporate out-of-domain data into the training process to improve the quality of the generated text.

2 Network

In all our experiments, we use our in-house attention-based sequence-to-sequence (seq2seq) implementation which is similar to Bahdanau et al. (2015). The approach is based on an encoder-decoder network. The encoder employs a bi-directional RNN to encode the input words $\mathbf{x} = (x_1, \dots, x_l)$ into a sequence of hidden states $\mathbf{h} = (h_1, \dots, h_l)$, where l is the length of the input sequence. Each h_i is a concatenation of a left-to-right \vec{h}_i and a right-to-left \overleftarrow{h}_i RNN:

$$h_i = \begin{bmatrix} \overleftarrow{h}_i \\ \vec{h}_i \end{bmatrix} = \begin{bmatrix} \overleftarrow{f}(x_i, \overleftarrow{h}_{i+1}) \\ \vec{f}(x_i, \vec{h}_{i-1}) \end{bmatrix}$$

where \overleftarrow{f} and \vec{f} are two gated recurrent units (GRU) proposed by Cho et al. (2014).

Given the encoded \mathbf{h} , the decoder predicts the target sentence by maximizing the conditional log-probability of the correct output $\mathbf{y}^* = (y_1^*, \dots, y_m^*)$, where m is the length of the target. At each time t , the probability of each word y_t from a target vocabulary V_y is:

$$p(y_t | \mathbf{h}, y_{t-1}^* \dots y_1^*) = g(s_t, y_{t-1}^*, H_t), \quad (1)$$

where g is a two layer feed-forward neural network over the embedding of the previous target word y_{t-1}^* , the hidden state s_t , and the weighted sum of \mathbf{h} (H_t).

Before we compute s_t and H_t , we first covert s_{t-1} and the embedding of y_{t-1}^* into an intermediate state s'_t with a GRU u as:

$$s'_t = u(s_{t-1}, y_{t-1}^*). \quad (2)$$

Then we have s_t as:

$$s_t = q(s'_t, H_t) \quad (3)$$

where q is a GRU, and the H_t is computed as:

$$H_t = \left[\begin{array}{c} \sum_{i=1}^l (\alpha_{t,i} \cdot \overleftarrow{h}_i) \\ \sum_{i=1}^l (\alpha_{t,i} \cdot \overrightarrow{h}_i) \end{array} \right], \quad (4)$$

The attention weights, α in H_t , are computed with a two layer feed-forward neural network r :

$$\alpha_{t,i} = \frac{\exp\{r(s'_t, h_i)\}}{\sum_{j=1}^l \exp\{r(s'_t, h_j)\}} \quad (5)$$

3 Unsupervised Approach

Our unsupervised model is based on the same training idea as a denoising auto-encoder (DAE) similar to Vincent et al. (2008). The original DAEs were feedforward nets applied to (image) data. In our experiments, the model architecture is a seq2seq model similar to Bahdanau et al. (2015). The idea of a DAE is to train a model that is able to reconstruct each training example from a partially destroyed input. This is done by first corrupting each training sequence x_i to get a partially destroyed version \tilde{x}_i .

In our unsupervised experiments, we generate the training data with the following corrupting process, parameterized by the desired percentage p of deletion: for each target sequence x_i , a fixed percentage p of words are removed at random, while the others are left untouched. We sample a new corrupt version \tilde{x}_i in each training epoch.

Instead of always removing a fixed percentage of words, we sample p for each sequence separately from a Gaussian distribution with mean $p = 0.6$ and variance 0.1. We chose $p = 0.6$ based on the average length ratio between the slot values and the target sequences in our labeled training data.

This corruption approach is motivated by the fact that many NLG problems are facing a similar task to the one the DAE is solving. Given some structured information, the task is to generate a target sequence that includes all the information. If we map the structured information to phrases that should be in the desired output, then the structured data problem resembles the DAE problem. For instance, if we have the following structured example: *name: Aromi - family friendly: yes* \rightarrow *Aromi has a family friendly atmosphere.*, we convert it into the input *Aromi family friendly* that we can feed to the DAE. To preprocess the structured data, we convert the boolean feature *family friendly* into a meaningful phrase ("family friendly") by using the slot name. For all non boolean slot pairs, we just use the slot values as meaningful phrases. Please keep in mind this transformation is only needed during inference as the training data has no slot pairs and only consists of pairs of corrupt and correct target sequences.

Nevertheless, there are two major differences between the training procedure of a DAE and an inference instance in NLG: First, we do not need to predict any content information in NLG as all of the content information is already provided by the structured data. On the other hand, a DAE training instance can also remove content words from the sentence. To align the two much closer, we restrict the words which the DAE is allowed to remove and apply the following heuristic to the corruption process of the DAE: Given the absolute counts $N(v_i)$ for each word v_i in our vocabulary, we only allow v_i to be removed when its count $N(v_i)$ is larger than a threshold. This heuristic is motivated by the fact that the corpus frequency of content words like a restaurant name is most likely low and the corpus frequency of non-content words like "the" is most likely high. The corpus frequencies can be either calculated on the training data itself or on a different corpora. The latter one has the advantage that domain specific content words that are frequent in the training data will have a low frequency in an out-of-domain corpora.

The second difference is that in a DAE training

	original	Loch Fyne is a family friendly restaurant providing Indian food .
(a)	remove random 60%	Fyne is restaurant food .
(b)	remove only words w_i with $N(w_i) > 100$	Loch Fyne family friendly Indian
(c)	shuffle words	family friendly Indian Loch Fyne

Table 2: Training data generation heuristics. (a): random 60% of the words are removed. (b): 60% of the words are removed, but only words that occur more than 100 times in the training data. Our assumption is that these are the non-content words. (c): On top of (b), all words are shuffled while keeping all word pairs (e.g. Loch Fyne) together that also occur in the original sentence.

instance, the words in a corrupt input occur in the same order as in the desired target. For an NLG inference instance, the order of the structured input does not need to match the order of the words in the output. To overcome this issue, we shuffle the words within the corrupt sentence while not splitting bigrams that also exist in the original sentence. An example of all three heuristics is given in Table 2.

4 Supervised Approach

For comparison, we train a supervised baseline based on the vanilla seq2seq model as described in Section 2. To make better use of the structured data, we found that the input word embeddings (wemb) of the seq2seq network should be represented together by the slot name and value. We split the word embedding vector into two parts and use the upper half for a word embedding of the slot name and the lower half for the word embedding of the slot value. If a slot value has multiple words, we build separate word embeddings for each word, but all having the same upper part (slot name). An example for the slot pairs of Table 1 is given in Figure 1.

Wemb (name)	Wemb (name)	Wemb (type)	Wemb (food)	Wemb (fam_ friend)
Wemb (Loch)	Wemb (Fyne)	Wemb (rest.)	Wemb (Indian)	Wemb (yes)

Figure 1: Example input word embeddings for our supervised baseline (Section 4) from the training Example of Table 1. The upper half of the word embedding is used for the slot names; the lower half for the slot values.

5 Data Sets

The **E2E data set** (Novikova et al., 2017) contains reviews in the restaurant domain. Given up to 8

different pieces of information about a restaurant, the task is to generate an English sentence that includes all of the provided structured information. The dataset comes with 42061 training examples. We split the provided dev set (547 examples) into validation (tail 268 examples) and test (head 279 examples), each having between 3 and 42 (on average 8) reference sentences. An example of an E2E training instance is given in Table 1.

The **news-commentary** data set is a parallel corpus of news provided by the WMT conference (Bojar et al., 2017) for training machine translation (MT) systems. For our unsupervised experiments, we use the English **news-commentary** part of the corpora only which contains 256,715 sentences.

All corpora are tokenized and we remove sentences that are longer than 60 tokens. In addition to tokenization, we also apply byte-pair encoding (Sennrich et al., 2016) when **news-commentary** is included in the setup.

dataset	task	# examples
E2E	NLG	42,061
news-commentary	MT (De-En)	256,715

Table 3: Training data statistics. Each training instance in NLG contains structured information and one reference sequence. Each training instance in MT contains one sentence written in both German and English.

6 Experiments

6.1 Model Parameters

For all of our experiments we utilize the seq2seq implementation as described in Section 2. We run inference with a beam size of 5. We use a hidden layer size of 1024 and a word embedding size of 620 and use SGD with an initial learning rate of 0.5. We halve the learning rate every other epoch starting from the 5th epoch. We evaluate the

generated text with BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), and NIST (Doddington, 2002) and use the evaluation tool provided by the E2E organizers to calculate the scores.

6.2 Automatic Scores

Our experimental results are summarized in Table 4. We list two supervised baselines: The first one is from the organizers of the E2E challenge, the second one is from our supervised setup (Section 4). Our baseline yields better performance on BLEU and ROUGE-L while reaching similar performance in NIST. Our third (unsupervised) *baseline copy input* just runs the evaluation metrics on the input (slot values of the structured data). This system performs much worse, but serves as a lower bound for our unsupervised experiments.

We report results on different unsupervised setups as described in Section 3. The system *randomly drop* just randomly drops 60% of the words, but still yields 57.3 BLEU, 65.9 ROUGE-L and 7.3 NIST points. You can easily detect a lot of extra information in the output that can not be explained by the structured input. Further, the output sounds very machine generated as the output depends on the order of the structured data. The heuristics *+ only words w/ count >100 (on ind data)* and *+ only words w/ count >100 (on ood data)* forbid removing words in the corruption phase that appear less than 100 times in the in-domain data or out-of-domain (ood) data, respectively. The latter setup uses the out-of-domain data for generating the word counts only and yields an improvement of 5.2 BLEU, 1.3 ROUGE-L and 0.2 NIST points compared to just randomly dropping words. The output still sounds very machine generated, but stops hallucinating additional information. We further improve the performance by 3.5 BLEU, 3.9 ROUGE-L, and 0.2 NIST points when shuffling the words in the corrupted input and using the out-of-domain data also as training examples. We use the English side of the 256,715 sentences from the news-commentary dataset as out-of-domain data only. We did not see any further improvements by adding more out-of-domain training data.

Finally, we build a semi supervised system that in addition to the unlabeled data includes the labeled information for some of the training examples. For these, we remove the slot names from the structured data and use a concatenation of all slot values as input to learn the correct output.

By jointly using both unlabeled and labeled data, we yield an additional improvement of 1.0 BLEU points compared to our best fully unsupervised system. In our semi supervised setup, we only use the slot values as input even for the labeled examples. This explains the drop in performance when comparing to the supervised setups. All supervised setups also include the slot names in their input representation.

6.3 Human Evaluation

In addition to automatic scores, we ran human assessment of the generated text as none of the automatic metrics correlates well with human judgment (Belz and Reiter, 2006).

To collect human rankings, we presented 3 outputs generated by 3 different systems side-by-side to crowd-workers, who were asked to score each sentence on a 6-point scale for:

- **fluency:** How do you judge the overall naturalness of the utterance in terms of its grammatical correctness and fluency?

For the next questions, we presented in addition to the 3 different system outputs, the structured representations of each example. We asked the crowd-worker to score the following two questions on a 5-point scale:

- **all information:** How much of the given information is mentioned in the text?
- **bad/ false information:** How much false or extra information is mentioned in the text?

Each task has been given to three different raters. Consequently, each output has a separate score for each question that is the average of 3 different ratings. The human evaluation results are summarized in Table 5. We included the two supervised baselines and our best unsupervised setup in the human evaluation. The unsupervised setup outperforms the supervised setups in fluency. One explanation is that our unsupervised system includes additional unlabeled data that can not be included in a supervised setup. Due to our unsupervised learning approach that all words in the structured data need to be included in the final output, the unsupervised system did not miss any information. Further, all three outputs included little false or wrong information that was not included in the structured data. All in all the output of the

setup	model	BLEU	ROUGE-L	NIST
supervised	baseline E2E challenge (Dušek and Jurcicek, 2016)	70.2	72.4	8.3
	baseline vanilla seq2seq (Section 4)	72.7	75.1	8.3
unsupervised	baseline copy input	27.7	56.4	3.2
	randomly drop	57.3	65.9	7.3
	+ only words w/ count >100 (on ind data)	59.5	66.4	7.3
	+ only words w/ count >100 (on ood data)	62.5	67.2	7.5
	+ shuffle pos	64.8	69.4	7.6
	+ ood data	66.0	71.1	7.7
semi-supervised	+ labeled ind data	67.0	72.1	7.8

Table 4: Results on the E2E dataset. The terms ood and ind are abbreviations for out-of-domain and in-domain respectively. The baseline systems as well as the unsupervised systems that are not labeled with + *ood data* are trained on the E2E in-domain training data only.

system	fluency	all information	extra/ false information
baseline E2E challenge (Dušek and Jurcicek, 2016)	4.01	4.89	0.05
baseline vanilla seq2seq (Section 4)	4.46	4.91	0.08
unsupervised (random drop + words w/ count >100 (ood data) + shuffle pos + ood data)	4.70 [†]	5.00 [†]	0.05

Table 5: Human evaluation results: We generated 279 output sequences for each of the 3 listed systems. Each sequence has been evaluated by 3 different raters and the score is the average of 837 ratings per system. For each task and sequence, the raters were asked to give a score between 0 and 5. A score of 5 for *fluency* means that the text is fluent and grammatical correct. A score of 5 for *all information* means that all information from the structured data is mentioned. A score of 0 for *extra/false information* means that no information besides the structured data is mentioned in the sequence. Scores labeled with [†] are significant better than all other systems ($p < 0.0001$).

unsupervised system is better than the two supervised systems.

We used approximate randomization (AR) as our significance test, as recommended by (Riezler and Maxwell, 2005). Pairwise tests between results in Table 5 showed that our novel unsupervised approach is significantly better than both baselines regarding fluency and mentioning all information with the likelihood of incorrectly rejecting the null hypothesis of $p < 0.0001$.

7 Limitations

Our unsupervised approach has two limitations and is therefore not easily applicable to all NLG problems or datasets. First, we can only run our approach for datasets where the input meaning representation either overlaps with target texts or we need to generate rules that map the structured data to target words. Unfortunately, the needed pattern can be very complicated and the effort of writing rules can be similar to the one of building a template based system.

Second, to be able to generate text from structured data during inference, the original structured input is converted to an unstructured one by discarding the slot names. This can be problematic in scenarios where the slot name itself contributes to the meaning representation, but the slot name should not be in the target text. For instances the structured data of a WEBNLG (Gardent et al., 2017) training example consists of several subject-predicate-object tuple features. Many of the features for one example have the same subject, but different predicates and objects. But yet in the final output, we prefer to have the subject only once.

8 Related Work

8.1 Neural Language Generation

Due to the recent success in Deep Learning, researchers started to use end-to-end systems to jointly model the traditional separated tasks of content selection, sentence planning and surface realization in one system. Recently, RNNs (Wen et al., 2015b), attention-based methods (Mei et al.,

2016) or LSTMs (Wen et al., 2015a) were successfully applied for the task of NLG. Liu et al. (2018) introduced a modified LSTM that adds a field gate into the LSTM to incorporate the structured data. Further, they used a dual attention mechanism that combines attention of both the slot names and the actual slot content. Sha et al. (2017) extended this approach and integrated a linked matrix in their model that learns the desired order of the slots in the target text. Further, Dušek and Jurcicek (2016) reranked the n-best output from a seq2seq model to penalize sentences that miss required information or add irrelevant ones. Instead of RNNs, Lebrecht et al. (2016) introduced a neural feed-forward language model conditioned on both the full structured data and the structured information of the previous generated words. In addition, the authors introduced a copy mechanism for boosting the words given by the structured data.

In contrast to the above mentioned related work, we train our model in a fully unsupervised fashion. Although, all our experiments have been conducted with the seq2seq model, our unsupervised approach can be applied on top of all of the different network architectures that are introduced by the above mentioned papers.

8.2 DAE and Unsupervised Learning

Denosing auto-encoders and unsupervised training have been applied to various other NLP tasks. Vincent et al. (2008) introduced denosing one-layer auto-encoders that are optimized to reconstruct input data from random corruption. The outputs of the intermediate layers of these denoisers are then used as input features for subsequent learning tasks such as supervised classification (Lee et al., 2009; Glorot et al., 2011). They showed that transforming data into DAE representations (as a pre-training or initialization step) gives more robust (supervised) classification performance. Lample et al. (2018) used a denosing auto-encoder to build an unsupervised Machine Translation model. Hill et al. (2016) trained a denosing auto-encoder on a seq2seq network architecture for training sentence and paragraph representations from the output of the intermediate layers. They showed that using noise in the encoder step is helpful to learn a better sentence representation.

In contrast to the above mentioned related work, we train a DAE directly on a task and do not

take the intermediate hidden states of a DAE as sentence representation to help learning a different task. Further, none of the related work applied DAEs on the task of generating sentences out of structured data. In addition, we modify the original DAE corruption process by introducing heuristics that remove non-content words only to match the input representation of a supervised NLG training instance.

9 Conclusion

We showed how to train a denosing auto-encoder that is able to generate correct English sentences from structured data. By applying several heuristics to the corruption phase of the auto-encoder, we reach better performance compared to two fully supervised systems. As no labeled data is needed in our approach, we further successfully improve the quality by incorporating out-of-domain data into the training phase. We run a human evaluation for the two supervised baselines and our best unsupervised setup. We see that the output of our unsupervised setup not only includes 100% of the structured information, but also outperforms both supervised baselines in terms of fluency and grammatical correctness.

The unsupervised training scheme gives us the option to incorporate any unlabeled data. One possible addition to our approach would be to incorporate text in different languages into our system, so that we can generate the output in any language from the same structured data.

Our approach is appropriate only for NLG problems where the goal is to include all the information from the structured data in the output. In future work, we will focus on the semi-supervised approach to make the DAE also suitable for problems where instead of all, only a subset of the structured information should be included in the output.

References

- D. Bahdanau, K. Cho, and Y. Bengio. 2015. *Neural Machine Translation by Jointly Learning to Align and Translate*. In *International Conference on Learning Representations (ICLR)*.
- Anja Belz and Ehud Reiter. 2006. *Comparing Automatic and Human Evaluation of NLG Systems*. In *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. [Findings of the 2017 Conference on Machine Translation \(WMT17\)](#). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. [On the Properties of Neural Machine Translation: Encoder-Decoder Approaches](#). *CoRR*, abs/1409.1259.
- 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 (HLT)*, pages 138–145. Morgan Kaufmann Publishers Inc.
- Ondřej Dušek and Filip Jurčíček. 2016. [Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 45–51.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. [Creating Training Corpora for Micro-Planners](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. [Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach](#). In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 513–520.
- Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. [Learning Distributed Representations of Sentences from Unlabelled Data](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1367–1377.
- Nal Kalchbrenner and Phil Blunsom. 2013. [Recurrent Continuous Translation Models](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1700–1709.
- Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018. [Unsupervised Machine Translation Using Monolingual Corpora Only](#). In *International Conference on Learning Representations (ICLR)*.
- Rémi Lebreton, David Grangier, and Michael Auli. 2016. [Neural Text Generation from Structured Data with Application to the Biography Domain](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1203–1213, Austin, Texas.
- Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. 2009. [Unsupervised feature learning for audio classification using convolutional deep belief networks](#). In *Advances in Neural Information Processing Systems (NIPS)*, pages 1096–1104.
- Chin-Yew Lin. 2004. [ROUGE: A Package for Automatic Evaluation of Summaries](#). In *Text Summarization Branches Out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.
- Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. [Table-to-text Generation by Structure-aware Seq2seq Learning](#). In *Association for the Advancement of Artificial Intelligence (AAAI)*.
- Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2016. [What to talk about and how? Selective Generation using LSTMs with Coarse-to-Fine Alignment](#). In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 720–730.
- Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. [The E2E Dataset: New Challenges For End-to-End Generation](#). In *Proceedings of the 18th Annual SIGDial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 201–206.
- 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 of the Association for Computational Linguistics (ACL)*, pages 311–318.
- Stefan Riezler and John T Maxwell. 2005. [On some pitfalls in automatic evaluation and significance testing for mt](#). In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 57–64.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural Machine Translation of Rare Words with Subword Units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 1715–1725.
- L. Sha, L. Mou, T. Liu, P. Poupart, S. Li, B. Chang, and Z. Sui. 2017. [Order-Planning Neural Text Generation From Structured Data](#). *ArXiv e-prints*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to Sequence Learning with Neural Networks](#). In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112.
- Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. [Extracting and composing robust features with denoising autoencoders](#). In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 1096–1103. ACM.

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 *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, page 275.

Tsung-Hsien Wen, Milica Gasic, 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)*, pages 1711–1721.