Reinforcement adaptation of an attention-based neural natural language generator for spoken dialogue systems

Matthieu Riou, Bassam Jabaian, Stéphane Huet, Fabrice Lefèvre

Dialogue & Discourse 10(1) (2019) 1–19 doi: 10.5087/dad.2019.101

Reinforcement adaptation of an attention-based neural natural language generator for spoken dialogue systems

Matthieu Riou MATTHIEU.RIOU@ALUMNI.UNIV-AVIGNON.FR
CERI-LIA, Avignon Université
Avignon, France

Bassam Jabaian BASSAM.JABAIAN@UNIV-AVIGNON.FR
CERI-LIA, Avignon Université
Avignon, France

Stéphane Huet STEPHANE.HUET@UNIV-AVIGNON.FR
CERI-LIA, Avignon Université
Avignon, France

Fabrice Lefèvre FABRICE.LEFEVRE@UNIV-AVIGNON.FR
CERI-LIA, Avignon Université
Avignon, France

Editor: Vera Demberg

Submitted 02/2018; Accepted 01/2019; Published online 02/2019

Abstract

Following some recent proposals to handle natural language generation in spoken dialogue systems with long short-term memory recurrent neural network models (Wen et al., 2016a), this work first investigates a variant thereof with the objective of a better integration of the attention sub-network. Our second objective is to propose and evaluate a framework to adapt the NLG module on-line through direct interactions with users. The basic way to do so is to lead the users to utter alternative sentences rephrasing the expression of a particular dialogue act. To add such a new sentence to its model, the system can rely on automatic transcription, which is costless but error-prone, or ask the user to transcribe it manually, which is almost flawless but costly. To optimise this choice, we investigate a reinforcement learning approach based on an adversarial bandit scheme. The bandit reward is defined as a linear combination of expected payoffs, on the one hand, and costs of acquiring the new data provided by the user, on the other hand. We show that this definition allows the system designer to find the right balance between improving the system performance, for a better match with the user's preferences, and limiting the burden associated with it. Finally, the actual benefits of the system are assessed through a human evaluation, showing that the progressive inclusion of more diverse utterances increases user satisfaction.

Keywords: natural language generation, recurrent neural network, adversarial bandit, on-line learning, user adaptation

1. Introduction

In a spoken dialogue system, the Natural Language Generation (NLG) component aims to produce an utterance from a system Dialogue Act (DA) decided by the dialogue manager. For instance, the

system DA: inform(name=bar_metropol, type=bar, area=north, food=french) may generate the utterance *Bar Metropol is a bar in the northern part of town serving French food*.

Traditional NLG systems use patterns and rules to generate system answers. Recently, several proposals have emerged to address the data-driven language generation issue (see for instance Rieser and Lemon, 2011, chap. 9) . They can be roughly grouped into two main categories: neural translation of dialogue acts and utterance language models.

In the latter group, generation is embedded into the whole process of interaction and each new system utterance is sampled from a neural network conditioned by the history of the dialogue (e.g., Serban et al., 2016) . In the first group, a more classical compositional approach has been followed, consisting in translating a targeted DA (or meaning representation) into a surface form (e.g., Wen et al., 2015b) with a recurrent network model close to the seq2seq model (Bahdanau et al., 2014). This work is in line with previous studies showing that the transfer between texts and DAs can be directly handled by a general language translation approach (Jabaian et al., 2016) or inverted semantic parsers (Konstas and Lapata, 2013).

In all these cases, a difficulty remains: a huge amount of data is required. We propose to address this difficulty hampering the practical development of such models by combining the current template-based approach with the on-line training of a neural NLG model. Some corpus extension methods are also possible (e.g., Manishina et al., 2016) but they do not allow a simultaneous adaptation to the user's preferences. The overall scheme consists in bootstrapping a first version of the model based on a corpus built with some simple templates and a small information database (to help fill in the template placeholders with values). This model sets up a first version of the dialogue system; once operational, the initial system is used to collect new training data while interacting with users.

It should be noted that at this critical step of development, users should still be under the control of the designers (they can be designers themselves or colleagues), as it can be hazardous to let the general public directly access such a functionality without any efficient means to counterbalance the effect of the on-line adaptation. This difficult and sensitive point will be addressed more thoroughly in future work.

The objective is to maintain the additional workload of the user resulting from the system's requests at an admissible level. Indeed, to collect new data for its model, the system will have to decide at each turn whether: 1. it should ask the user for an alternative to its current answer, 2. it can use the automatic transcription of the user's input directly or ask for additional processing. Basically, such processing will consist in manual corrections of the transcription, but ideally this step could also be handled vocally, which could be rather tedious if done properly.

This paper is organised as follows: after presenting related work in Section 2, we define our novel NLG model in Section 3. Section 4 describes the framework we propose to adapt the model on-line through direct interactions. Section 5 provides an experimental study with automatic and human evaluations of our approach. We conclude our discussion and propose further perspectives in Section 6.

2. Related work

Template-based models still constitute the mainstream method used in the NLG field for commercial purposes. They rely on hand-crafted rules and linguistic resources and turn out to produce good-quality utterances for repetitive and specific tasks (Rambow et al., 2001). For this reason,

the NLG component has long received less attention in dialogue system research than spoken language understanding or dialogue management components for instance. However, recent studies have tried to alleviate two main drawbacks of the template-based models: the lack of scalability to large open domains and the frequent repetition of identical and mechanical utterances (see Gatt and Krahmer, 2018, for a recent survey of the current trends in the NLG field) . One example is to build upon stylistic generation with psychological underpinnings to adjust to the user's personality dynamically (Mairesse and Walker, 2010).

Data-driven and stochastic approaches have been devised to increase maintainability and extensibility. Oh and Rudnicky proposed to use a set of word-based n-gram Language Models (LMs) to over-generate a set of candidate utterances, from which the final form is selected (Oh and Rudnicky, 2002). Mairesse and Young extended this model by introducing factors built over a coarse-grained semantic representation to build phrase-based LMs (Mairesse and Young, 2014). More recently, Wen et al. have proposed several models based on Recurrent Neural Networks (RNNs) (Wen et al., 2015a,c,b; Mei et al., 2016). Some recent extensions include the proposition of Dušek and Jurčíček of a SEQ2SEQ model with attention to produce both strings and deep syntax trees in a joint generation, replacing the classical pipeline (Dušek and Jurčíček, 2016).

Evaluations made by human judges show that these systems are able to generate high-quality utterances which are also more linguistically varied than template-based approaches. The use of recurrent encoder-decoder NNs has also been investigated to build end-to-end dialogue systems in a non-goal-directed context, for which large corpora are available (Serban et al., 2016), or selective generation from weather forecasting and sportscasting datasets (Mei et al., 2016).

Our proposal is related to two of the generation models proposed by Wen et al.: 1. the Semantically Conditioned LSTM-based model (SCLSTM) introduces an additional control cell into the Long Short-Term Memory (LSTM) to decide for each generated word what information to retain for the remaining part of the utterance (Wen et al., 2015a); 2. the RNN encoder-decoder architecture with an attention mechanism encodes the dialogue act into a distributed vector representation with attention screening over slot-value pairs updated after each generated word. After that, a decoder eventually produces a word sequence with an LSTM network (Wen et al., 2015b).

Stochastic models still require extensive work to produce corpora for new domains. Novikova et al. proposed a crowd-sourcing framework to collect data for NLG (Novikova et al., 2016). Wen et al. presented an incremental recipe to deal with the domain adaptation problem for RNN-based generation models (Wen et al., 2016b). They used counterfeited data synthesised from an out-of-domain dataset to fine-tune their model on a small set of in-domain utterances.

We still aim at reducing the burden to produce new data, not to adapt to another domain like Walker et al. (2007), but to generate more diverse utterances better adapted to the user's preferences. To this end, a reinforcement learning approach based on an adversarial bandit scheme is applied (Auer et al., 2002). This approach has been used previously in dialogue systems for language understanding (Ferreira et al., 2015, 2016). Here, we propose a protocol to adapt the RNN-based model on new utterances that vary from the training dataset, taking into account the cost for the user to provide these examples.

Other approaches based on active learning have been used in NLG. For instance, Mairesse et al. (2010) included an active learning protocol in their language generator in order to optimise the data collection process, using a model which can determine the next semantic input to annotate based on its estimated certainty about the correctness of its output. Likewise, Fang et al. (2017) proposed a deep reinforcement learning algorithm capable of learning an active learning strategy from data in

order to decide whether or not to annotate each utterance. But our work is the first to propose the use of an adversarial bandit algorithm to support the decision-making process for the active learning in an NLG data collection. The use of bandit algorithms has already been investigated in various active learning protocols, for example in recommendation systems by Li et al. (2010), or in dialogue systems where they have been applied to automatically update spoken language understanding models deployed in a spoken service that evolved with time (Gotab et al., 2009).

3. A Combined-Context LSTM for language generation

This section presents the generation model proposed in this paper. It is based on two previous models: the Semantically Conditioned LSTM (Wen et al., 2015a) and the Attention-based RNN Encoder-Decoder (Wen et al., 2015b). After a detailed description, the proposed model is compared with the reference models to point out precisely where the expected benefits of the combined model lie. Then the training and decoding processes are described.

3.1 Model description

Our model is based on the same recurrent neural architecture as (Wen et al., 2015a). The overall principle is to generate each new element of the word sequence conditioned on the previously-generated one, a hidden (recurrently updated) vector and a context-information vector. In practice, a 1-hot encoding $\mathbf{w_{t-1}}$ of a token[1] w_{t-1} is input to the model at each time step t conditioned on a recurrent hidden layer $\mathbf{h_{t-1}}$, from which the probability distribution of the next token w_t is defined. To ensure that the generated utterance represents the intended meaning, an additional context vector $\mathbf{d_t}$, encoding the dialogue act and its associated slot-value pairs, is also provided at each step t.

As in the attention-based encoder-decoder, the decoding process is performed through a standard LSTM (Fig. 1b), which is fed by an additional vector $\mathbf{a_t}$ representing the information on which the model currently focuses (Fig. 1a). $\mathbf{a_t}$ is called the local DA embedding with attention. The set of relations between all the vectors involved in the LSTM cell is:

$$\mathbf{i}_t = \text{sigmoid}(\mathbf{W}_{wi}\mathbf{w}_{t-1} + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ai}\mathbf{a}_t) \quad (1)$$

$$\mathbf{f}_t = \text{sigmoid}(\mathbf{W}_{wf}\mathbf{w}_{t-1} + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{af}\mathbf{a}_t) \quad (2)$$

$$\mathbf{o}_t = \text{sigmoid}(\mathbf{W}_{wo}\mathbf{w}_{t-1} + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{ao}\mathbf{a}_t) \quad (3)$$

$$\hat{\mathbf{c}}_t = \tanh\left(\mathbf{W}_{wc}\mathbf{w}_{t-1} + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{W}_{ac}\mathbf{a}_t\right) \quad (4)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right) \quad (6)$$

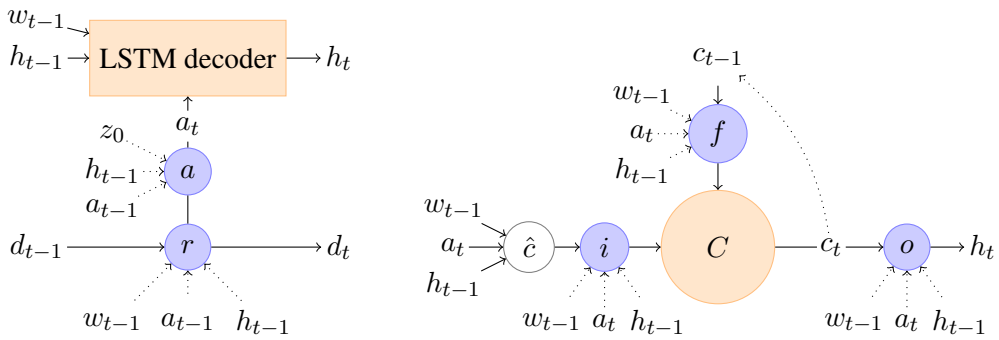
where $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t \in [0,1]^n$ are input, forget and output gates respectively, $\hat{\mathbf{c}}_t$ and $\mathbf{c}_t$ are proposed and true cell values at time t, and $\odot$ denotes element-wise multiplication.

Subsequently, the next token w_t is picked up, either through argmax or sampling, on the output distribution formed as:

$$P(w_t|w_{t-1}, w_{t-2}, ...w_0, \mathbf{a}_t) = \text{softmax}(\mathbf{W}_{ho}\mathbf{h}_t) \quad (7)$$

$$w_t \sim P(w_t|w_{t-1}, w_{t-2}, ...w_0, \mathbf{a}_t) \quad (8)$$

1. The same terminology as in Wen et al. (2015a) is used since the input text is also delexicalised: the slot values (e.g."Chinese" for slot food) are replaced in the input by their corresponding slot tokens (e.g. SLOT_FOOD).



(a) Global architecture of the Combined-Context LSTM.

(b) Details of the LSTM decoder.

Figure 1: Combined-Context LSTM

where $\mathbf{W}_{ho}$ is the output weight matrix.

As with the attention-based encoder-decoder model, the local DA embedding $\mathbf{a}_t$ is computed from the global DA embedding $\mathbf{d}_t$, which represents the information remaining to express in the rest of the generation. To define the initial global DA $\mathbf{d}_0$, each slot-value pair is embedded as a vector $\mathbf{z}_{0,i}$:

$$\mathbf{z}_{0,i} = \mathbf{s}_i + \mathbf{v}_i \tag{9}$$

where $\mathbf{s}_i$ and $\mathbf{v}_i$ are the i-th slot and value pair of the dialogue act, each represented by a 1-hot representation. Then the complete dialogue act is represented by:

$$\mathbf{d}_0 = \mathbf{act}_0 \oplus \sum_i \mathbf{z}_{0,i} \tag{10}$$

where $\mathbf{act}_0$ is a 1-hot representation of the act type and $\oplus$ stands for vector concatenation. Therefore, the global dialogue act $\mathbf{d}_t$, corresponding to the remaining information to deliver, is obtained at each time step by:

$$\mathbf{d}_t = \mathbf{act}_t \oplus \sum_i \mathbf{z}_{t,i} \quad . \tag{11}$$

$\mathbf{act}_t$ and $\mathbf{z}_{t,i}$ are updated with the parts of $\mathbf{s}_i$ and $\mathbf{v}_i$ that remain in the global dialogue act $\mathbf{d}_t$ according to the reading gate r (Fig. 1a):

$$\mathbf{d}_t = \mathbf{r}_t \odot \mathbf{d}_{t-1} \tag{12}$$

$$\mathbf{r}_t = \text{sigmoid}(\mathbf{W}_{wr}\mathbf{w}_{t-1} + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{W}_{ar}\mathbf{a}_{t-1}) \quad . \tag{13}$$

The local DA embedding $\mathbf{a}_t$, representing the information we want to focus on at step t, is formed as:

$$\mathbf{a}_t = \mathbf{act}_t \oplus \sum_i \omega_{t,i}\mathbf{z}_{t,i} \tag{14}$$

where $\omega_{t,i}$ is the weight of i-th slot-value pair computed by the attention mechanism a:

$$\omega_{t,i} = \text{softmax}(\beta_{t,i}) \tag{15}$$

$$\beta_{t,i} = \mathbf{q}^{\mathsf{T}}.\tanh\left(\mathbf{W}_{hm}\mathbf{h}_{t-1} + \mathbf{W}_{mm}\mathbf{z}_{0,i} + \mathbf{W}_{am}\mathbf{a}_{t-1}\right) \tag{16}$$

$\mathbf{q}$ and $\mathbf{W}$s being parameters to learn.

3.2 Comparison with the reference models

The generation model proposed here combines the Semantically Conditioned LSTM (Wen et al., 2015a) and the attention-based RNN Encoder-Decoder (Wen et al., 2015b). Each of these models proposes a way to process the semantic information represented as a DA to produce an utterance. Without delving into details (for which we strongly advise to refer to the original papers), we briefly recall the structure of the two models in Figure 2 and try to summarise their main differences w.r.t. the processing of their input data, the dialogue acts.

The SCLSTM reading-gate handles the DA by choosing what information to retain or discard at each step as illustrated in Figure 2 (a). For this purpose, at each step the reading-gate takes as input the last turn's remaining unprocessed information $\mathbf{d}_{t-1}$ and outputs $\mathbf{d}_t$, conditioned on the previous word $\mathbf{w}_{t-1}$ and the LSTM state $\mathbf{h}_{t-1}$.

Conversely, the attention mechanism takes as input the initial DA $\mathbf{d}_0$ at each step, and outputs the information to process $\mathbf{a}_t$, conditioned on the initial DA $\mathbf{d}_0$ and the previous LSTM state $\mathbf{h}_{t-1}$. However, it loses the progression of unprocessed information (Figure 2 (b)). Subsequent to this, for both models, an LSTM decoder generates the next LSTM state $\mathbf{h}_t$ from which the next word $\mathbf{w}_t$ is picked up (using Equation 7), conditioned on the previous word $\mathbf{w}_{t-1}$ and the LSTM state $\mathbf{h}_{t-1}$.

Each model offers some advantages and inconveniences. A plus is that the SCLSTM is less inclined to forget slots as the reading gate informs on the remaining unprocessed information. A disadvantage is that it tends to deliver incoherent and ungrammatical sentences in order to deliver all the slots at all costs. For example, for the following input DA:

inform(name=restaurant ducroix, kids_allowed=no, phone=4153917195, postcode=94111, address=690 sacramento street)

an SCLSTM generates:

*The address of restaurant ducroix is 690 sacramento street child and **allowed** and **is 4153917195** and the postcode is 94111.*

where concepts kids_allowed and phone are present but wrongly formulated.

For its part, the encoder-decoder uses the attention mechanism to select the part of the DA that should be considered by the LSTM decoder at each step. Thus the system can better process each slot locally. But it has no dedicated mechanism to ensure that all slots have been processed at the end of sentence. For example, for the following DA:

inform(name=thep phanom thai, address=400 waller street, postcode=94117, phone=415431256)

an attention-based RNN Encoder-Decoder proposes:

The address for thep phanom thai restaurant is 400 waller street and the postcode is 94117.

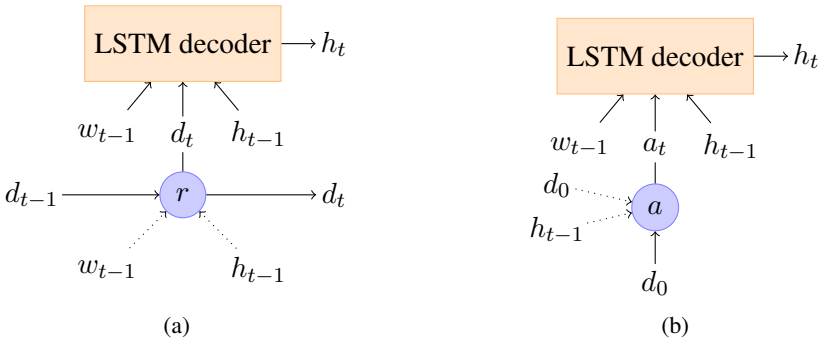


Figure 2: Semantically Conditioned LSTM (a) and Attention-based RNN Encoder-Decoder (b).

which is grammatically correct but in which the phone number is missing.

Our objective is to combine the advantages of both models, using a reading gate and an attention mechanism to sequentially process the DA. Thus, the system is less inclined to forget or misprocess some slots, and as a consequence should improve its BLEU score and slot error rate.

Therefore, in practice, the major novelty of our model lies in the computation of $\mathbf{a}_t$ in the attention mechanism (see Equation 14), which takes as input the current DA $\mathbf{d}_t$ (decomposed in $\mathbf{act}_t$ and $\mathbf{z}_{t,i}$) instead of the initial DA $\mathbf{d}_0$. This current DA is obtained from the output of a reading-gate $\mathbf{r}_t$ (see Equations 12 and 13). Besides, the previous attention's output $\mathbf{a}_{t-1}$ is added as a parameter in both the reading-gate $\mathbf{r}_t$ and the attention's weights $\omega_{t,i}$ (see Equations 15 and 16).

3.3 Training and decoding

The objective function used to train the weights of the network computes the cross-entropy between the predicted token distribution $\mathbf{p}_t$ and the actual token distribution $\mathbf{y}_t$:

$$F(\theta) = \sum_t (\mathbf{p}_t^{\mathsf{T}} \log(\mathbf{y}_t)) + \|\mathbf{d}_T\| + \sum_{t=0}^{T-1} (\eta * \xi^{\|\mathbf{d}_{t+1} - \mathbf{d}_t\|}) \quad . \tag{17}$$

Following Wen et al. (2015c), an l_2 regularisation term is introduced as well as a second regularisation term[2] required to control the reading gate dynamics. We optimise the parameters with stochastic gradient descent and back propagation through time. Early-stopping on a validation set prevents over-fitting.

The decoding is split into two steps: 1. during an over-generation phase, the system is used to generate several utterances for the given DA, by randomly picking the next token on the output distribution, and 2. in a subsequent re-ranking phase, each utterance is ranked on the basis of a score R calculated as:

$$R = -(F(\theta) + \lambda \text{ERR}) \tag{18}$$

where λ is a trade-off constant, set to 10, and ERR is the slot error rate. $ERR = (p + q)/N$ with N the total number of DA slots, and p, q the number of missing and redundant slots in the proposed utterance, compared to the input DA.

2. T is the total number of steps, $\eta = 10^{-4}$, $\xi = 100$.

4. On-line interactive problem

Neural NLG can give good results, but it requires a large amount of annotated data to be trained in order to have an efficient model with diversity in its outputs. Several examples of utterances for each DA are then required to train the model. In order to reduce the cost of collecting such a corpus, the following on-line learning protocol was set up.

We propose to proceed in two main steps:

1. a bootstrap corpus, consisting of references generated from templates, is used to train a generation model;
2. this learned model generates new utterances and the users are required to propose better or varied alternatives.

In order to reduce the effort on the user's side and to avoid useless actions, we propose to rely on an adversarial bandit algorithm to decide whether the system should prod the user considering the expected gain and cost of its action or not.

4.1 Static case

Once the system generates the utterance, the system can choose one action (from a probability distribution) among a set $\mathcal{I}$ of M actions. In this preliminary setup, we consider a case where $M = 3$ and $\mathcal{I}$ can be defined as:

$$\mathcal{I} := \{\text{Skip, AskDictation, AskTranscription}\}.$$

Let $i \in \mathcal{I}$ be the action index. We assume that the user effort $\phi(i) \in \mathbb{N}$ can be measured by the time needed to perform action i. The *actions* and associated *user efforts* are:

- **Skip**: skip the refinement process. The cost of this action is always set to 0 ($\phi(\text{skip}) = 0$).
- **AskDictation**: refine the model, taking into account an alternative utterance proposed by the user and transcribed automatically with an ASR system ($\phi(\text{AskDictation}) = 1$).
- **AskTranscription**: ask the user to transcribe the correction or the alternative utterance. Two different costs are considered for this action:
  - Un-normalised cost: $\phi(\text{AskTranscription}) = 1 + l$
  - Normalised cost: $\phi(\text{AskTranscription}) = 1 + \frac{l}{L_{max}}$

  with l the length of the proposed utterance, and L_{max} the maximum possible length (set to 40 words in our experiments).

Then the *gain* of the chosen action is estimated as follows:

- **Skip**: nothing is learned, gain is 0 ($g(\text{skip}) = 0$).
- **AskDictation**: we propose to compute the gain as the remaining margin of the BLEU-4 score that would have been obtained by the utterance generated by the system, using the user-proposed utterance as a reference, noted $\text{BLEU}_{gen/prop}$. To take into account the potential errors added by the ASR system, the gain is penalised by the estimations of WER and ERR:

$$g(\text{AskDictation}) = (1 - \text{BLEU}_{gen/prop}) \times (1 - \text{WER}) \times (1 - \text{ERR}) \ .$$

The global WER expresses the confidence we can have in the BLEU-4 measure (as it is based on erroneous utterances), while the slot error rate ERR penalises utterances that do not contain the required semantic information, due to ASR errors.

- **AskTranscription**: asking the user to manually transcribe the utterance prevents ASR errors. Therefore, the gain estimate only considers the BLEU-4 score of the utterance generated by the system, using the user-proposed sentence as a reference ($g(\text{AskTranscription}) = 1 - \text{BLEU}_{gen/prop}$).

Finally, a loss function is defined $l(i) \in [0, 1]$ such that the system, through an optimisation process, seeks to maximise the gain measure $g(i)$ and to minimise the user effort $\phi(i)$ jointly:

$$l(i) = \underbrace{\alpha(1 - g(i))}_{\text{system improvement}} + \underbrace{(1 - \alpha)\frac{\phi(i)}{\phi_{max}}}_{\text{user effort}} \tag{19}$$

Very importantly α allows to weight the payoff w.r.t. the cost, allowing the designer to influence the system's behavior depending on the targeted operational conditions (from fast improvement, no matter the cost, down to slow improvement to preserve users' efforts).

4.2 Adversarial bandit case

The following scenario for the adversarial bandit problem is considered: the system produces a sentence then chooses an action $i_t \in \mathcal{I}$. Once the action i_t is performed, the system computes: (a) the gain estimate $g_t(i_t)$ with the user collaboration, (b) the user effort $\phi_t(i_t)$ and (c) the current loss.

The goal of the bandit algorithm is then to find $i_1, i_2, \ldots$, so that for each T, the system minimises the total loss as expressed in the previous section.

Every n iterations, the user-proposed utterances are added to the training corpus, and the model is updated on this extended corpus. At the same time, we compute the loss function for each bandit's choice, and update its policy.

5. Experimental study

In this section, the improvement of the Combined-Context LSTM over SCLSTM and Encoder-decoder is measured (Section 5.1). Then the on-line learning protocol is evaluated on simulated data in Section 5.2. In order to evaluate whether (or not) the on-line learning approach has an impact on real users' subjective appreciation of systems, a human evaluation is made in Section 5.3. Finally, in Section 5.4, the impact of the WER simulation is evaluated on a smaller dataset, collected with a real ASR.

5.1 System comparison

A first set of experiments was conducted on the SF restaurant corpus, described in Wen et al. (2015c) and freely accessible.[3] It contains 5 191 utterances, for 271 distinct DAs. With each DA, the corpus associates a template-generated utterance and several utterances in natural English proposed by humans, each utterance being delexicalised.

3. https://www.repository.cam.ac.uk/handle/1810/251304

System	BLEU-4	ERR (%)
SCLSTM	**0.722***	0.66
Encoder-decoder	0.697	0.65
Combined-Context LSTM	0.711	**0.24****

* Significant w.r.t the Encoder-decoder by the t-test (p-value< 0.01)

** Significant by the t-test (p-values< 0.001)

Table 1: Results on the top 5 hypotheses.

Candidate \ Reference	SCLSTM	Encoder-decoder	Combined-Context LSTM
SCLSTM	-	0.857	0.883
Encoder-decoder	0.847	-	0.824
Combined-Context LSTM	0.849	0.840	-

Table 2: BLEU-4 cross-comparison of the three systems.

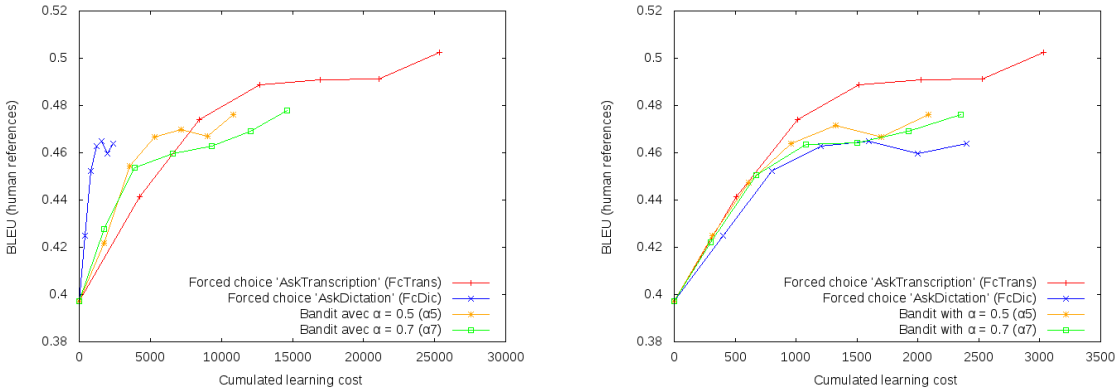
Our model and both the SCLSTM and the Attention-based RNN Encoder-Decoder were implemented using the Tensorflow library[4] and were trained on a corpus split into 3 parts: training, validation and testing (3:1:1 ratio), using only the human-proposed utterance references.

The three systems were compared using two metrics: the BLEU-4 score (Papineni et al., 2002) and the slot error rate (ERR). The BLEU-4 value validates the utterance generation, especially grammaticality, but is often not seen as a useful measure of content quality for NLG (Reiter and Belz, 2009). To remedy that deficiency, ERR, which concentrates only on the semantic contents but with more accuracy, is also computed. For each example, we over-generated 20 utterances and kept the top 5 hypotheses for evaluation. Each hypothesis has been processed as an independent output sentence to evaluate, and so averaged during the BLEU-4 computation. Multiple references for each DA were obtained by grouping delexicalised utterances with the same DA specification, and then "relexicalised" with the proper values.

As can be seen in Table 1, the BLEU-4 score of the Combined-Context LSTM falls between the two other systems (roughly 0.01 gap between each) but the measured differences were not statistically significant (p-value> 0.01 for each pair of systems,[5] except between the SCLSTM and the Encoder-decoder with a p-value$= 0.002$). However, the slot error rate is reduced by one third by our new model w.r.t. the two other systems, the improvement being statistically significant between SCLSTM and Combined-Context LSTM (p-value< 0.001). This means that, while the new model does not really achieve to learn more diverse responses, it offers a better coverage of the expressed concepts resulting in fewer omitted concepts, which is the first purpose of an NLG system.

To evaluate the differences in sentence generation, BLEU-4 has been used to compare the outputs of each system with the two others. As can be seen from Table 2, all cross-system BLEU-4 scores are pretty high, around 0.80-0.90. This indicates that systems tend to produce comparable sentences, and as a consequence we did not investigate further the possibility of combining these different neural models into one.

4. `https://www.tensorflow.org`
5. Statistical significance was computed using a two-tailed Student's t-test between each pair of systems.



(a) Un-normalised cost

(b) Normalised cost

Figure 3: Evolution of BLEU-4 score, as a function of the cumulated learning cost.

5.2 On-line adaptation evaluation

For the evaluation of the on-line adaptation procedure, the same corpus is used again. But this time, training, validation and testing parts follow a 2:1:1 ratio (to maintain a test set of minimal size despite a smaller corpus). The Combined-Context system is used to train an initial bootstrap model on the training set, using the template-generated utterance references. The validation corpus was used for early stopping, again with the template-generated references. Then, we simulated the on-line learning on the same training set, using this time the enclosed human-proposed references.

The model and bandit updates were learned every 400 utterances. WER was simulated by randomly inserting errors (confusion, deletion, insertion) into the corpus examples until a pre-defined global WER was reached. This WER simulation put aside the idiosyncratic properties of the used language and ASR system. But realistic models are very complex to develop and never really satisfactory, notably because different ASRs make different errors. For this reason, a rather simple random model for simulation has been chosen, and a test was carried out with an actual ASR system afterwards in the user trials. The models are trained on delexicalised utterances, which allows the computation of ERR. We note that the ERR score can be reduced when the value of the slot-value pair does not appear in the surface form of utterances (for example with the "dont_care" value). In the on-line learning setup, it can raise new issues if a user proposed an alternative surface form which does not contain the wanted value, resulting in a higher ERR score.

The initial model, trained on the template-generated part of the training corpus, obtains a high BLEU-4 score, 0.802, when tested on the template-generated part of the test, but this value is dramatically reduced to 0.397 on the human-proposed references. This tends to confirm that even a well-trained model does not compete with the diversity of possible responses occurring in a conversation in natural language.

Figure 3 plots the BLEU-4 score as a function of the learning cost, the simulated WER being set to 5%. BLEU-4 is obtained by testing the model against the human-proposed subset of the test. The learning cost is computed as the sum of the costs of all choices made by the bandit during the learning. Different configurations are tested: the forced 'AskDictation' choice (FcDic) and the forced 'AskTranscription' choice (FcTrans). Besides, the bandit is tested with two α values: 0.5

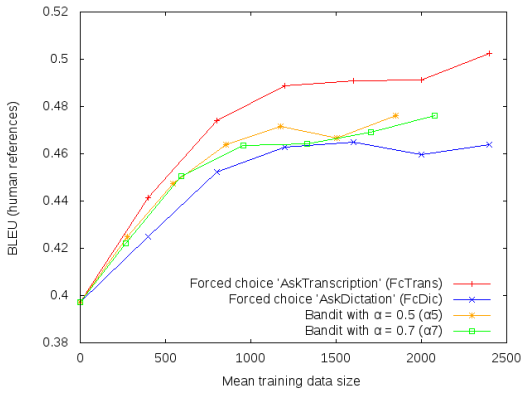


Figure 4: Evolution of BLEU-4 score, as a function of the training data size.

(α5) and 0.7 (α7), each displayed with normalised and un-normalised costs. The second value reduces the influence of the cost, allowing the system to increase the effort asked to the user. Each curve is composed of seven points; the first one corresponds to the score of the system before on-line learning, the six others are computed after each block of 400 utterances. The cost is cumulative over all previous blocks.

With the un-normalised cost (Figure 3a), we can observe that the bandit succeeds in reducing the cost of learning up to a certain amount of training data (after a cumulated cost of 7 500, Ask-Transcription outperforms all other configurations). After using all the training data, α5 and α7 reach both 0.476 BLEU, an intermediate value between 0.464 for FcDic and 0.503 for FcTrans. AskTranscription costs much more than AskDictation, therefore, at first, the bandit learns better than FcTrans by balancing between the two choices. But after the first two blocks, the increase reduces until both the α5 and α7 curves pass below FcTrans. A higher α value tends to favour the Ask actions over Skip, and AskTranscription over AskDictation.

The normalised cost has been tested with the same configuration. The results (Figure 3b) are quite similar. AskTranscription still outperforms all other configuration and each setup reaches the same BLEU-4 score as with the un-normalised cost. Nevertheless, the cost is much lower for α5 (2 085), α7 (2 348) and AskTranscription (3 034). The normalised cost reduces the gap between the estimated costs of a dictation and a transcription; thus, it tends to favour more the AskTranscription.

Figure 4 displays the BLEU-4 scores as a direct function of the training data size. The results are consistent with previous conclusions, since both α5 and α7 achieve their learning with a reduced amount of data, and performances between the forced Ask choices.

The bandit was also tested with a higher WER (20%). At this rate, the system no longer learns from the choice AskDictation, errors overwhelming improvement. The forced choice AskDictation gives a BLEU-4 score of 0.383, less than the initial system. An analysis of the learned policy shows that with a low WER (5%), the bandit globally explores both Ask learning choices, and presents at the end a slight preference for AskDictation. With a higher WER (20%), the bandit favours AskTranscription (chosen almost half of the time at the last iteration), due to more utterances with a high slot error rate and therefore a lower gain.

5.3 Human evaluation

Objective evaluations using automatic metrics like BLEU-4 do not necessarily reflect real users' preferences (Callison-Burch et al., 2006). In particular, naturalness is really hard to formalise (as in general, more socially-oriented qualities of a system cannot be easily captured with current evaluation modes (e.g., Curry et al. (2017) or Perez-Beltrachini and Gardent (2017))

In order to better evaluate whether or not the results of the on-line learning approach induce a better appreciation of the system by the users, a human evaluation was conducted. Any of the enhanced models could have been used as the objective here is to confirm that the observations in the simulated environment (BLEU and EER scores) transferred well to subjective user's appraisal, that is: Does the adaptation procedure, whatever the exact model used, improve the quality of the generation process? However, due to the cost of the experiment, only the adapted model $\alpha 5$ was chosen to be compared with the initial model (only referred to as 'the adapted model' hereafter).

Five annotators were recruited to automatically evaluate generated utterances. For each example, a dialogue act and the three best sentences generated by each system have been presented to the annotators. The six utterances were randomly ordered, and there was no indication on the system that built them. The annotators have been asked to give three scores to each utterance, all rating from 1 to 3:

- **Informativeness** evaluates whether the information given by the dialogue act is well expressed in the generated utterance:
  - 3: all information given by the dialogue act is conveyed and no additional information is introduced;
  - 2: minor information is missing or there is extra information not present in the dialogue act;
  - 1: any other case.
- **Syntax** evaluates the level of syntactic correctness of an utterance:
  - 3: the utterance is correct;
  - 2: there are small, hardly audible imperfections;
  - 1: there are some clear mistakes in the utterance.
- **Naturalness** evaluates whether the utterance sounds like a potential human production:
  - 3: the utterance could have been pronounced by a human in this situation;
  - 2: the utterance is correct but unfit to the situation, or sounds synthetic;
  - 1: even by correcting the grammatical errors if needed, the utterance could never have been pronounced by a human.

To reduce the burden of the annotators, duplicate sentences were merged. In addition to the evaluation of each sentence, they were asked to indicate their preferred one. To evaluate annotator agreement using the Fleiss Kappa metrics, the first 20 examples were shared over all annotators.

A total of 471 evaluations were collected. The κ for the global annotator agreement is 0.550. The task on which the annotators less agreed is rating naturalness, with a κ of 0.468 to be compared with 0.594 and 0.576 respectively for informativeness and syntax correctness.

	Initial system	Adapted system
Global score	2.356	**2.425***
Informativeness	**2.528**	2.509
Syntax	2.272	**2.383***
Naturalness	2.267	**2.383***

* $p < 0.001$

Table 3: Average scores for each system. Statistical significance was computed using a two-tailed Student's t-test between the two systems.

Act type	System	All	Informativeness	Naturalness	Syntax
inform	Initial	2.50	2.39	2.39	2.42
	Adapted	2.42	2.52	2.37	2.39
inform_only_match	Initial	2.33	2.50	2.50	2.00
	Adapted	1.94	2.00	2.00	1.83
?inform_no_match	Initial	2.48	2.78	2.33	2.34
	Adapted	2.19	2.23	2.07	2.05
?select	Initial	2.16	2.52	2.01	1.89
	Adapted	2.05	2.52	2.37	2.33
?request	Initial	2.65	2.82	2.65	2.47
	Adapted	2.63	2.77	2.57	2.55
?reqmore	Initial	2.27	2.67	2.07	2.07
	Adapted	2.62	2.47	2.73	2.67
?confirm	Initial	2.02	2.27	1.91	1.88
	Adapted	2.05	2.33	1.94	1.88
goodbye	Initial	1.71	1.70	1.70	1.72
	Adapted	2.59	2.60	2.59	2.57

Table 4: Average scores for each system w.r.t. the act type of the dialogue act.

As shown in Table 3, the adapted system obtains a significantly higher global average score than the initial system. More specifically, the adapted system obtains significantly higher scores for naturalness and syntax, and the slight decrease in informativeness is not significant.

Table 4 indicates the variation of scores w.r.t. act type. Contrary to what one might think, it can be observed that the scores are quite regular over all act types, event though they each represent very variable levels of complexity. On the contrary, Table 5 shows that both systems tend to have higher overall scores for dialogue acts of low or medium length (2 or 3 slots). With more slots, the scores tend to decrease as the utterances become more complicated and more conducive to errors with automatic generation.

When the annotators were asked to vote for their favourite utterance, they mainly voted for the best utterance of each system, with a significant preference for the adapted system (Table 6). But mostly the second and third best propositions of the adapted systems were also selected quite often, unlike the second and third best propositions of the initial system. This tends to confirm that the adapted system can generate more satisfying sentences with a variability greater than the initial system.

# slots	System	All	Informativeness	Naturalness	Syntax
0 slot	Initial	1.74	1.76	1.72	1.74
	Adapted	2.59	2.60	2.60	2.58
1 slot	Initial	2.38	2.59	2.34	2.21
	Adapted	2.38	2.55	2.31	2.27
2 slots	Initial	2.58	2.80	2.50	2.46
	Adapted	2.64	2.75	2.58	2.58
3 slots	Initial	2.47	2.72	2.30	2.39
	Adapted	2.32	2.48	2.26	2.29
4 slots	Initial	2.28	2.35	2.24	2.27
	Adapted	1.71	1.74	1.69	1.70
5 slots	Initial	1.75	2.00	1.67	1.58
	Adapted	1.66	1.50	1.75	1.58

Table 5: Average scores for each system w.r.t. the number of slots in the dialogue act.

Rank	Initial system	Adapted system
1	111 (22.0%)	143 (28.5%)
2	51 (10.2%)	103 (20.6%)
3	18 (3.6%)	75 (15.0%)
Total	180 (35.9%)	**321 (64.1%)***

* $p < 0.001$

Table 6: Number of selected sentences by annotators w.r.t. their rank. Statistical significance was computed by means of a two-tailed binomial test.

5.4 On-line adaptation using real ASR data evaluation

To evaluate in practice the on-line framework, and in particular the impact of the WER during the interactions, a data collection has been carried out with true users.

To collect the data, a dialogue act with some possible references was presented to the user for each example. Then, the user has been asked to dictate an alternative sentence corresponding to the dialogue act. To facilitate the experiment deployment, the capabilities of speech recognition available in the browser Chrome, with the Google ASR API, were used. With the sentence transcribed, the user had the possibility to manually correct the automatic output if needed. Both the transcribed and corrected utterances were collected, as well as the confidence score of the automatically transcribed sentences.

426 pairs of transcriptions (automatic, manual) have been collected this way.[6] The transcribed utterances present a mean confidence score of 0.86 and a mean WER (between transcribed and corrected utterances) of only 2.42%.

A new model was adapted, through the on-line adaptation experimentation described in section 5.2 using the new collected data instead of the former human-proposed references. The new corpus was divided in 300 examples for training and 126 for testing. The same initial bootstrap model was used, but this time it was updated, as well as the bandit, every 50 utterances due to the

6. All data used in this study are available upon request to the authors.

smaller size of the corpus. To enhance the gain estimation, the estimated WER was replaced by the confidence score of the transcription:

$$g(\text{AskDictation}) = (1 - \text{BLEU}_{gen/prop}) \times \text{confidence score} \times (1 - \text{ERR}) \ .$$

This new adapted model has been tested on the same corpus as for the first experiment, by comparing the generated utterances to the references generated by patterns, and to the references proposed by human annotators including the corrected references of the new collected oral data. Results are similar to the experiments done with the simulated WER. When using patterns as references, BLEU-4 is reduced by 0.10 (from 0.829 to 0.727), while it slightly decreases by 0.03 (from 0.482 to 0.451) when compared to human references. The lowest scores observed with this last setup with human references can be explained by the large number of human utterances from the initial corpus that have not been learned in this experiment.

The bandit algorithm avoids having the system interfere too often with the user. During the entire learning, it asked for transcriptions 53% of the time, compared to only 23% for dictation, and it did not ask anything (skipped) 23% of the time. In this way, the cumulated cost of the learning has been divided by two w.r.t. a system that would always ask for transcription (from 4 243 to 2 430) without decreasing too much learning performance (BLEU-4 score is 0.458 for the forced transcription system against 0.444 with the bandit). However, it has a lower performance than a system that would always ask for dictation, which obtains a BLEU-4 score of 0.451 for a cumulated cost of 300.

To allow the system to better estimate whether it has to ask or not for an oral alternative, and whether or not a transcription is needed, the context has to be taking into account by the bandit (the nature of the dialogue act, complexity...). This could be done with the same protocol by using a contextual bandit (Auer et al., 2002) instead of the adversarial bandit. However, such a setup is very likely to converge faster than the non-contextual version and thus limit the exploration steps, which the adversarial bandit maintains steadily. In any case, a comparison of the variants is planned to obtain more insights on their true performance.

6. Conclusion

In this paper we have investigated an attention-based neural network for natural language generation, combining two systems proposed by Wen et al.: the Semantically Conditioned LSTM-based model (SCLSTM) and the RNN encoder-decoder architecture with an attention mechanism. While not improving the BLEU score globally, this model outperforms them on the slot error rate, preventing the semantic repetitions or omissions in the generated utterances. We then proposed a protocol to adapt a bootstrapped model using on-line learning. Results obtained on a simulated experiment have been confirmed and completed with real users, providing new proposals to the system and assessing the qualities of the adapted system's hypotheses. The bandit algorithm has been shown to allow the system to balance between improving the system's performance and the additional workload it implies for the user. It also leads to a system considered more varied by the users. In future work, we will study how to improve the system's learning ability by taking the context into account before making a choice, with recourse to a contextual bandit. More importantly, the natural language generator has to be evaluated within an entire dialogue system to definitely confirm the practical interest of the overall approach.

7. Acknowledgements

This work has been partially carried out within the Labex BLRI (ANR-11-LABX-0036).

References

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. sep 2014. URL `http://arxiv.org/abs/1409.0473`.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of BLEU in machine translation research. In *EACL*, pages 249–256, 2006.

Amanda Cercas Curry, Helen Hastie, and Verena Rieser. A review of evaluation techniques for social dialogue systems. In *Proceedings of the 1st ACM SIGCHI International Workshop on Investigating Social Interactions with Artificial Agents*, ISIAA 2017, pages 25–26, 2017.

Ondřej Dušek and Filip Jurčíček. 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 (Volume 2: Short Papers)*, pages 45–51, Berlin, Germany, 2016.

Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, 2017.

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefèvre. Zero-shot semantic parser for spoken language understanding. In *Proceedings of INTERSPEECH*, 2015.

Emmanuel Ferreira, Alexandre Reiffers-Masson, Bassam Jabaian, and Fabrice Lefèvre. Adversarial bandit for online interactive active learning of zero-shot spoken language understanding. In *Proceedings of ICASSP*, 2016.

Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018.

Pierre Gotab, Frédéric Béchet, and Géraldine Damnati. Active learning for rule-based and corpus-based spoken language understanding models. In *IEEE Workshop on Automatic Speech Recognition & Understanding, 2009, ASRU 2009*, pages 444–449. IEEE, 2009.

Bassam Jabaian, Fabrice Lefèvre, and Laurent Besacier. A unified framework for translation and understanding allowing discriminative joint decoding for multilingual speech semantic interpretation. *Computer Speech & Language*, 35:185–199, 2016.

Ioanis Konstas and Mirella Lapata. A global model for concept-to-text generation. *Journal of Artificial Intelligence Research*, 48:305–346, 2013.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference*

on World Wide Web, WWW '10, pages 661–670, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772758. URL http://doi.acm.org/10.1145/1772690.1772758.

François Mairesse and Marilyn A. Walker. Towards personality-based user adaptation: Psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, 20 (3):227–278, August 2010.

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1552–1561, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1858681.1858838.

François Mairesse and Steve Young. Stochastic language generation in dialogue using factored language models. *Computational Linguistics*, 40(4):763–799, 2014.

Elena Manishina, Bassam Jabaian, Stéphane Huet, and Fabrice Lefèvre. Automatic corpus extension for data-driven natural language generation. In *Proceedings of LREC*, May 2016.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. What to talk about and how? Selective generation using LSTMs with coarse-to-fine alignment. In *Proceedings of NAACL-HLT*, 2016.

Jekaterina Novikova, Oliver Lemon, and Verena Rieser. Crowd-sourcing NLG data: Pictures elicit better data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 265–273. Association for Computational Linguistics, 2016.

Alice H. Oh and Alexander I Rudnicky. Stochastic natural language generation for spoken dialog systems. *Computer Speech & Language*, 16(3–4):387–407, 2002.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002.

Laura Perez-Beltrachini and Claire Gardent. Analysing data-to-text generation benchmarks. In *Proceedings of the 10th International Natural Language Generation conference*, Santiago de Compostelle, Spain, September 2017.

Owen Rambow, Srinivas Bangalore, and Marilyn Walker. Natural language generation in dialog systems. In *Proceedings of HLT*, 2001.

Ehud Reiter and Anja Belz. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558, 2009.

Verena Rieser and Oliver Lemon. *Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation*. Theory and Applications of Natural Language Processing. Springer-Verlag New York Inc, 2011.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2016.

Marilyn Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research*, 30(1): 413–456, November 2007.

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

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. Toward multi-domain language generation using recurrent neural networks. In *Proceedings of NIPS Workshop on Machine Learning for Spoken Language Understanding and Interaction*, 2015b.

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

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. Technical report, University of Cambridge, 2016a. URL `https://arxiv.org/abs/1604.04562`.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of NAACL-HLT*, 2016b.