Online adaptation of an attention-based neural network for natural language generation

Matthieu Riou, Bassam Jabaian, Stéphane Huet, Fabrice Lefèvre

Online adaptation of an attention-based neural network for natural language generation

Matthieu Riou, Bassam Jabaian, Stéphane Huet and Fabrice Lefèvre

CERI-LIA, University of Avignon, France

matthieu.riou@alumni.univ-avignon.fr, {bassam.jabaian, stephane.huet, fabrice.lefevre}@univ-avignon.fr

Abstract

Following some recent propositions to handle natural language generation in spoken dialog systems with long short-term memory recurrent neural network models [1] we first investigate a variant thereof with the objective of a better integration of the attention subnetwork. Then our main objective is to propose and evaluate a framework to adapt the NLG module online through direct interactions with the users. When doing so the basic way is to ask the user to utter an alternative sentence to express a particular dialog act. But then the system has to decide between using an automatic transcription or to ask for a manual transcription. To do so a reinforcement learning approach based on an adversarial bandit scheme is retained. We show that by defining appropriately the rewards as a linear combination of expected payoffs and costs of acquiring the new data provided by the user, a system design can balance between improving the system's performance towards a better match with the user's preferences and the burden associated with it.

Index Terms: natural language generation, recurrent neural network, adversarial bandit, online 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) following the dialogue manager decision. For instance, the system dialog act: $inform(name = bar\_metropol, type = bar, area = north, food = french)$ may generate the utterance "*Bar Metropol is a bar in the northern part of town serving French food*."

Traditional systems use patterns and rules to generate system answers. Recently several propositions have emerged to address the data-driven language generation issue. They can be grouped into two main categories: neural translation of dialog acts and utterance language models.

In this last group, generation is embedded in the whole process of interaction. And a new utterance is sampled from a neural network conditioned on the history of the dialog (e.g. [2]). Whereas in the first group, colleagues have followed the preceding compositional approach consisting in translating a targeted dialog act (or meaning representation) into a surface form (e.g. [3, 4]) with a recurrent network model close to the seq2seq model [5], in accordance with some previous studies showing that the transfer between texts and dialog acts can be directly handled by a general language translation approach [6].

In all cases a difficulty is the need for a huge amount of data. We address this difficulty hampering practical development of such models by combining the current template-based approach with an online training of a neural NLG model. Some corpus extension methods are also possible (e.g. [7]) but they do not allow a simultaneous adaptation to the user's preferences. The overall scheme is to bootstrap a first version of the model based on a corpus elaborated by means of the templates and a small information database. Such a model should allow to setup a first version of the dialog system. Once operational the initial system is used to collect new training data while interacting with users. In this critical step of development users may still be under control of the designers (they can be engineers themselves or colleagues), as it can be hazardous to let the general public directly access such a functionality without efficient mean to counter-balance the effect of the online adaptation. This difficult point will be addressed in a future work though. The principle is to maintain the additional workload of the user due to the system requests at an admissible level. Indeed to collect new data for its model, the system will have to decide at each turn if it 1. should ask the user an alternative to its answer, 2. can use the transcription of the user's input directly or ask for supplementary processing (basically corrections of transcription, but ideally this step is also handled vocally and so can be rather tedious to be done safely).

The reminder of this paper is organized 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 online through direct interactions. Section 5 provides evaluation using objective metrics. We conclude our discussion and propose ongoing avenues in Section 6.

2. Related work

Template-based models are still the mainstream method used in the natural language generation field. They rely on hand-crafted rules and linguistic resources and turn out to produce utterances of good quality for repetitive and specific tasks [8]. For this reason, the natural language generation component has long been received less attention in the dialogue system research domain than Spoken Language Understanding (SLU) or Dialogue Management components. However, recent studies 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.

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 LMs to overgenerate a set of candidate utterances, from which the final form is selected [9]. Mairesse and Young extended this model by introducing factors built over a coarse-grained semantic representation to build phrase-based LMs [10]. More recently, Wen et al. have proposed several models based on RNNs [11, 12, 4]. Evaluations made by human judges show that these systems are able to generate utterances with a high quality and more linguistically varied than template-based systems. The use of recurrent encoder-decoder NNs has also been investigated to build end-

to-end dialogue systems [2]. In this framework, RNNs carry out NLG but also natural language understanding and decision-making; the study focuses on non-goal-driven systems, for which large corpora are available.

In this paper, we propose a new model that combines two of the generation models proposed by Wen et al.: the Semantically Conditioned LSTM-based model (SCLSTM) introduces into the LSTM an additional control cell to decide for each generated word what information to retain for the next words [11]; the RNN encoder-decoder architecture with the attention mechanism encodes the dialogue act into a distributed vector representation with an attention over slot-value pairs updated for each generated word, then a decoder produces a word sequence with a LSTM network [4].

Stochastic models still require an extensive work to produce corpora for new domains. Wen et al. present an incremental recipe to deal with the domain adaptation problem for RNN-based generation model [13]. They resorted to counterfeited data synthesised from an out-of-domain dataset to fine-tune their model on a small set of in-domain utterances. In this article, we still aim at reducing the burden to produce new data, although not to adapt to another domain but to generate more diverse utterances. In this respect, a reinforcement learning approach based on an adversarial bandit scheme is applied [14]. If this approach has previously been used in dialogue systems for language understanding [15, 16], we propose a protocol to adapt the RNN-based model on new utterances that vary from the training dataset, taking into account the cost it implies for the user to give these examples.

3. A Combined-Context LSTM for language generation

The generation model proposed in this paper combines the Semantically Conditioned LSTM and the Attention-based RNN Encoder-Decoder. Each of these systems proposes a way to treat the semantic information represented as a DA to produce an utterance. The SCLSTM reading-gate handles the DA by choosing at each step which information to retain or to discard in future steps. Therefore the reading gate outputs at each step the remaining untreated information. On the contrary, the attention mechanism outputs the information to treat at the current step, but loses the progression of untreated information. The purpose of our system is to combine the advantages of both systems, using a reading gate and an attention mechanism to sequentially treat the DA. The reading gate aims at retaining the untreated information, while the attention mechanism selects the part of the DA that should be considered by the LSTM decoder at the current step.

3.1. System description

Like in previous RNN architectures for NLG [11], a 1-hot encoding $\mathbf{w_{t-1}}$ of a token[1] w_{t-1} is input to our model at each time step t conditioned on a recurrent hidden layer $\mathbf{h_{t-1}}$ and outputs the probability distribution of the next token w_t. To ensure that the generated utterance represents the intended meaning, an additional vector encoding $\mathbf{d_t}$ representing the dialogue act and its associated slot-value pairs is input at each step t.

Like the attention-based encoder-decoder [4], decoding is made by a standard LSTM, which is fed by an additional vector

[1] We use the same terminology as in [11] since the input text is also delexicalised: the slot values (e.g."Chinese food") are replaced by their corresponding slot tokens (e.g. SLOT_FOOD).

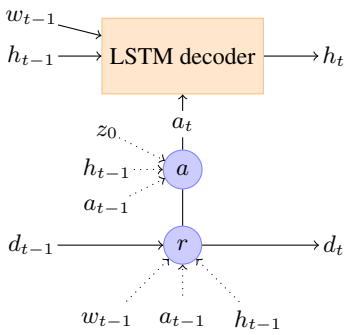

Figure 1: *Combined-Context LSTM*

$\mathbf{a_t}$ representing the information we currently want to express (Fig. 1). $\mathbf{a_t}$ is called the local DA embedding with attention. The next token w_t is picked up on the output distribution formed by:

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

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

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

The local DA embedding $\mathbf{a}_t$ is computed from the global DA embedding $\mathbf{d}_t$, representing the information remaining to express in the remainder of the generation. To represent the initial global DA $\mathbf{d}_0$, we embed each slot-value pair as a vector representation $\mathbf{z}_{0,i}$:

$$\mathbf{z}_{0,i} = \mathbf{s}_i + \mathbf{v}_i \quad (3)$$

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

$$\mathbf{d}_0 = \mathbf{act}_0 \oplus \sum_i \mathbf{z}_{0,i} \quad (4)$$

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

$$\mathbf{d}_t = \mathbf{act}_t \oplus \sum_i \mathbf{z}_{t,i} \quad . \quad (5)$$

$\mathbf{act}_t$ and $\mathbf{z}_{t,i}$ are updated according to the reading gate r (Fig. 1) similarly to the SCLSTM model:

$$\mathbf{d}_t = \mathbf{r}_t \odot \mathbf{d}_{t-1} \quad (6)$$

$$\mathbf{r}_t = \text{sigmoid}(\mathbf{W}_{wr}\mathbf{w}_{t-1} + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{W}_{ar}\mathbf{a}_{t-1}) \quad (7)$$

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

$$\mathbf{a}_t = \mathbf{act}_t \oplus \sum_i \omega_{t,i}\mathbf{z}_{t,i} \quad (8)$$

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

$$\omega_{t,i} = \text{softmax}(\beta_{t,i}) \quad (9)$$

$$\beta_{t,i} = \mathbf{q}^{\mathsf{T}}.\tanh\left(\mathbf{W}_{hm}\mathbf{h}_{t-1} + \mathbf{W}_{mm}\mathbf{z}_{0,i} + \mathbf{W}_{am}\mathbf{a}_{t-1}\right) \quad (10)$$

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

3.2. Training and decoding

The objective function computes the cross-entropy between the predicted token distribution $\mathbf{p}_t$ and the actual token distribution $\mathbf{y}_t$:

$$F(\theta) = \sum_t (\mathbf{p}_t^{\mathsf{T}} \log(\mathbf{y}_t)) + \|\mathbf{d}_T\| + \sum_{t=0}^{T-1} (\eta * \xi^{\|\mathbf{d}_{t+1} - \mathbf{d}_t\|}) \ . \tag{11}$$

Following [12], an l_2 regularisation term is introduced as well as a further regularisation[2] required to control the reading gate dynamics. We optimise the parameters with stochastic gradient descent and back propagation through time. In order to prevent over-fitting, we used early-stopping on a validation set.

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

$$R = -(F(\theta) + \lambda \text{ERR}) \tag{12}$$

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

4. Online interactive problem

Neural NLG can give good results, but it requires a large amount of annotated data to be trained in order to have a good performing model presenting diversity in the outputs. Several examples of utterances for each DA is then required to train the model. In order to reduce the cost of collecting such a corpus, an online learning protocol is proposed in this paper.

We propose to proceed in two steps : 1. a bootstrapping corpus, consisted of template-generated references, is used to train a generation model, then 2. this learned system generates utterances and asks the user for better or different ways to answer. In order to reduce the effort from the user side and avoid useless actions, we propose to rely on an adversarial bandit algorithm to decide whether it should ask the user or not considering the expected gain and cost of its action.

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 we defined are:

• **Skip**: Skip the refinement process. The cost of this action is always set to 0 ($\phi(\text{skip}) = 0$).

• **AskDictation**: Refine the model by considering an alternative utterance proposed by the user and transcribed automatically by an automatic system ($\phi(\text{AskDictation}) = 1$).

• **AskTranscription**: Ask the user to transcribe the correction or the alternative utterance.

We considered two different costs for this action:

- Unnormalized cost: $\phi(\text{AskTranscription}) = 1 + l$
- Normalized cost: $\phi(\text{AskTranscription}) = 1 + \frac{l}{L_{max}}$

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

We then estimate the gain of the chosen action, as follows:

• **Skip**: Nothing is learned, gain is 0 ($g(\text{skip}) = 0$).

• **AskDictation**: We compute the gain as the remaining margin of the BLEU 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 with the global estimate WER of the ASR system and the ERR:

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

The global WER expresses the confidence we have in the BLEU 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 from ASR errors. Therefore, the gain estimate only considers the BLEU score of the utterance generated by the system, using the user-proposed sentence as reference ($g(\text{AskTranscription}) = 1 - \text{BLEU}_{gen/prop}$).

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

$$l(i) = \underbrace{\alpha(1 - g(i))}_{\text{system improvement}} + \underbrace{(1-\alpha)\frac{\phi(i)}{\phi_{max}}}_{\text{user effort}} \tag{13}$$

α weights the payoff w.r.t. the cost, allowing the system to adapt to the user's preferences.

4.2. Adversarial bandit case

We consider the following scenario for adversarial bandit problem: 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 collaboration of the user, (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, we add the user-proposed utterances to our training corpus, and update the model 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

5.1. Evaluation setup

The experiments have been conducted on the SF restaurant corpus described in [12], and freely accessible.[3] It contains 5 191 utterances, for 271 distinct DAs. The corpus associates with each DA a template-generated utterance and several utterances in natural English proposed by humans, each utterance being delexicalised.

We implemented our system and both the SC-LSTM and the Attention-based RNN Encoder-Decoder using the Tensorflow library.[4] Then we trained them on a corpus split into 3

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

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

[4] https://www.tensorflow.org

Table 1: *Results on the top 5 hypotheses*

System	BLEU (%)	ERR (%)
SCLSTM	**72.2**	0.77
Encoder-decoder	69.7	0.65
Combined Context LSTM	71.1	**0.24**

parts: training, validation and testing (3:1:1 ratio), using only the human-proposed utterance references.

We compared the three systems using two metrics, the BLEU-4 score [17] and the slot error rate (ERR). The BLEU value validates the utterance generation, especially the grammaticality, while the ERR concentrates only on the semantic contents but with more accuracy. For each example, we over-generated 20 utterances and kept the top 5 hypotheses for evaluation. Multiple references for each DA were obtained by grouping delexicalised utterances with the same DA specification, and then "relexicalised" with the proper values.

5.2. System comparison

As can be seen in Table 1, the BLEU score of the Combined Context LSTM falls between the two other systems (roughly 1% gap between each), but the slot error rate is reduced by one third, w.r.t. the two other systems. It means that, while it 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 a NLG system.

5.3. Online adaptation evaluation

The same corpus is used again, but this time training, validation, testing parts follow a 2:1:1 ratio. We used our system to train an initial bootstrap model on the training set, using the template-generated utterance references. The validation corpus was used to do early stopping, again with the template-generated references. Then, we simulated the online learning on the same training set, using this time the human-proposed references. The model and bandit updates were learned every 400 utterances. The WER was simulated, by randomly inserting errors (confusion, deletion, insertion) into the corpus examples until we reached a pre-defined global WER.

The initial model, trained on the template-generated part of the training corpus, offers a high BLEU score, 80.2%, when tested on the template-generated part of the test, but only 39.7% on the human-proposed references. Even a well-trained model does not compete with the diversity of possible responses in a conversation in natural language.

Figure 2 plots the BLEU score as a function of the learning cost, the WER being set to 5%. The BLEU score is obtained by testing the model on the human-proposed part of the test. The learning cost is computed as the sum of the costs of all choices made by the bandit during the learning. Results are provided with unnormalized costs but some preliminary experiments showed that cost normalization has no relevant effect on the overall process. Different configurations are tested: the forced 'AskDictation' choice (FcDic) and the forced 'AskTranscription' choice (FcTrans). Besides, we tested the bandit with two α values: 0.5 ($\alpha 5$) and 0.7 ($\alpha 7$). 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 five points. The first one corresponds to the score of the system before online learning. The four others are computed after each block of 400 utterances. The cost is cumulative over all previous blocks.

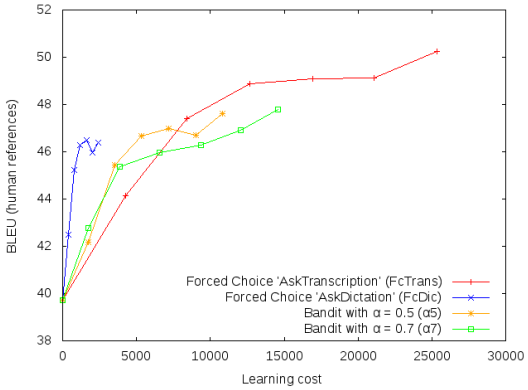

Figure 2: *Evolution of BLEU score, as a function of the cumulated learning cost.*

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, AskTranscription outperforms all other configurations). After using all the training data, $\alpha 5$ and $\alpha 7$ reach both 47.6% BLEU, intermediate between 46.4% for FcDic and 50.3% for FcTrans. AskTranscription costs much more than AskDictation, therefore, at first, the bandit learns better than FcTrans by balancing between the two choices. But passed the first two blocks, the increase reduces until both the $\alpha 5$ and $\alpha 7$ curves pass below FcTrans. A higher α value tends to favour the Ask actions over Skip, and the AskTranscription over the AskDictation.

We also tested the bandit with a higher WER (20%). At this rate, the system does not longer learn from the choice AskDictation, the errors overwhelming the improvement. The forced choice AskDictation gives a BLEU score of 38.3%, 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 50% of the time at the last iteration), due to more utterances with a high slot error rate and therefore a very low gain.

6. Conclusions

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. Then, we proposed a protocol to adapt a bootstrapped model using online learning. A bandit algorithm has been shown to allow the system to balance between improving the system's performance w.r.t. the cost it implies for the user. In a future work, a real setup will be designed to study how to improve the system's learning ability by taking into account the context before making a choice, with recourse to a contextual bandit [14].

7. Acknowledgements

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

8. References

[1] T.-H. Wen, M. Gašić, N. Mrkšić, L. M. R. Barahona, P.-H. Su, S. Ultes, D. Vandyke, and S. Young, "Conditional generation and snapshot learning in neural dialogue systems," in *EMNLP*, 2016.

[2] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *AAAI Conference on Artificial Intelligence*, 2016.

[3] T.-H. Wen, M. Gašić, N. Mrkšić, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, D. Vandyke, and S. Young, "A network-based end-to-end trainable task-oriented dialogue system," University of Cambridge, Tech. Rep., 2016. [Online]. Available: https://arxiv.org/abs/1604.04562

[4] T.-H. Wen, M. Gašić, N. Mrkšić, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, and S. Young, "Toward multi-domain language generation using recurrent neural networks," in *NIPS Workshop on Machine Learning for Spoken Language Understanding and Interaction*, 2015.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv preprint arXiv:1409.0473*, Sep. 2014.

[6] B. Jabaian, F. Lefèvre, and L. Besacier, "A unified framework for translation and understanding allowing discriminative joint decoding for multilingual speech semantic interpretation," *Computer Speech & Language*, vol. 35, pp. 185–199, 2016.

[7] E. Manishina, B. Jabaian, S. Huet, and F. Lefèvre, "Automatic corpus extension for data-driven natural language generation," in *LREC*, May 2016.

[8] O. Rambow, S. Bangalore, and M. Walker, "Natural language generation in dialog systems," in *HLT*, 2001.

[9] A. H. Oh and A. I. Rudnicky, "Stochastic natural language generation for spoken dialog systems," *Computer Speech & Language*, vol. 16, no. 3–4, pp. 387–407, 2002.

[10] F. Mairesse and S. Young, "Stochastic language generation in dialogue using factored language models," *Computational Linguistics*, vol. 40, no. 4, pp. 763–799, 2014.

[11] T.-H. Wen, M. Gašić, D. Kim, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young, "Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking," in *SIGDIAL*, 2015.

[12] T.-H. Wen, M. Gašić, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young, "Semantically conditioned LSTM-based natural language generation for spoken dialogue systems," in *EMNLP*, 2015.

[13] T.-H. Wen, M. Gašić, N. Mrkšić, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, and S. Young, "Multi-domain neural network language generation for spoken dialogue systems," in *NAACL-HLT*, 2016.

[14] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[15] E. Ferreira, B. Jabaian, and F. Lefèvre, "Zero-shot semantic parser for spoken language understanding," in *INTERSPEECH*, 2015.

[16] E. Ferreira, A. Reiffers-Masson, B. Jabaian, and F. Lefèvre, "Adversarial bandit for online interactive active learning of zero-shot spoken language understanding," in *ICASSP*, 2016.

[17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002.