An Unsupervised Approach to Automatic Response Generation for Conversational e-Commerce Agents using Monte Carlo Tree Search

Snehasish Mukherjee
Stanford University
mukherji@stanford.edu

Abstract

Conversational e-commerce agents often generate responses based on templates that are instantiated using slots. Such systems are therefore unable to entertain the users way of speaking and end up sounding robotic. We are interested in generating entertaining natural language responses to user queries when we already know the user intent and the factual elements of the answer, i.e. slots. We model the task as a stochastic search problem over the space of all possible sentences that includes the slot and can be generated from the users query. We represent the state as an ordered sequence of words and specify permutation and transform operators to carry out the state transition. We propose a new scoring function designed to reward grammatical correctness and proximity to the users way of speaking. We show that our state transition operators and scoring function lead the MCTS to generate natural responses for small to medium length queries, while the performance for long queries needs further tuning.

1. Introduction

1.1. Motivation

The e-commerce industry has moved very fast on enabling voice commerce with all the major players having released some form of conversational shopping capability. Recent advances in machine learning, especially deep learning, has endowed conversational e-commerce agents with profound intent and named entity recognition capabilities. This allows such conversational systems to correctly detect the same intent irrespective of the way the user speaks. For example, a properly trained NLU system will identify both "What is there in my cart" and "What do I have in my cart" as the *queryCart* intent. However these advances are yet to reflect on the responses from these agents, which are mostly templatized. As shown in Figure 1, in both these cases the response is usually something like "Your cart contains an apple". The responses do not take into account the subtle differences in user's choice of words. In other words responses generated using fixed templates cannot entertain the users way of speaking. This is not only a major user-experience problem, user studies have also shown that inflexible robotic responses do not inspire trust and users are unlikely to return resulting in poor adoption of the technology. In this project we explore if we can alleviate this problem by automatically generating responses in an unsupervised manner using reinforcement learning so that the responses easily adapt to the users way of speaking thereby appearing more attentive and natural.

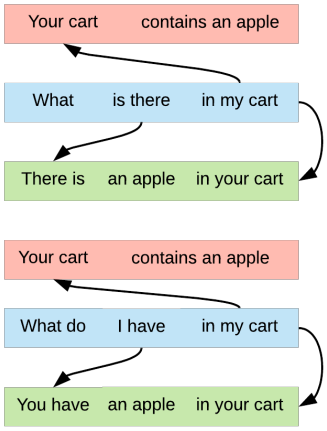

Figure 1. The blue question can be replied with the green entertaining answer but instead we have the red

1.2. Setup

Conversational systems usually have the structure as shown in Figure 1 The intent and NER module handles complexities in natural language very well and hence understands what the user wants. Accordingly the intent executor executes the intent and generates certain words or phrases, also called slots, that form a part of the answer, (i.e. "ginger bread", "great value" and "$2.49" are the slots in Figure 2).

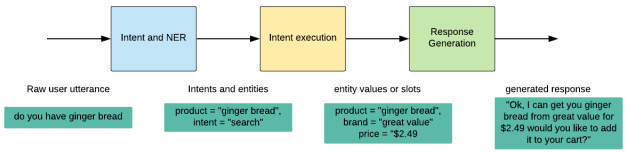

Figure 2. Execution flow of a conversational grocery assistant

The last module, which is the response generator, generates the final response by instantiating some templates using these slots. In Figure 2 the template could have been "*Ok, I can get you ${product} from ${brand} for ${price}. Would you like to add it to your cart?"*. We will only try to fix this last module, by automatically generating the responses instead of instantiating a fixed template. It may be noted here that **we are not trying to generate the answer to the user's question (that is done by the second module), but given the answer, we are trying to generate a sentence that will include the answer** and sound more personalized to the users way of speaking, also called "entertaining" in NLG literature.

2. Problem Definition

Given the original user utterance denoted as an ordered sequence of words

$$u : \{u_1, u_2, ...u_n\} \tag{1}$$

and the answer slot a which we treat to be atomic, we want to generate a response $\mathcal{R}$ given by

$$\mathcal{R} \quad = \quad \mathcal{R}_p(u)\ a\ \mathcal{R}_s(u) \tag{2}$$

where the prefix $\mathcal{R}_p(u)$ and the suffix $\mathcal{R}_s(u)$ are, possibly empty, ordered sequences of words, that depend on the original utterance u. Both $\mathcal{R}_p(u)$ and $\mathcal{R}_s(u)$ are to be computed jointly to that the final response $\mathcal{R}$ satisfies the following qualitative constraints

1. $\mathcal{R}$ is a valid English sentence (it is grammatically correct)
2. $\mathcal{R}$ has high score as determined by a conversational language model (it makes sense),
3. $\mathcal{R}$ does not sound like a question, and
4. $\mathcal{R}$ sounds very similar to what the user said.

As a concrete example, given "u = $\{$*what do i have in my cart*$\}$" and the answer slot "a = *an apple*", we want to generate "$\mathcal{R}$ = $\{$*you have an apple in your cart*$\}$" where $\mathcal{R}_p$ = $\{$*you have*$\}$ and $\mathcal{R}_s$ = $\{$*in your cart*$\}$

3. Related Work

Natural Language Generation is an active area of research. In [11] Lemon models NLG as a Markov Decision Process (MDP), but instead of focusing on the sentence generation, the paper tries to solve the problem of choosing the right template (i.e List, Contrast, Cluster etc) to present the information. In [18] Swanson et.al discusses a generative language modelling approach to NLG when all the words are constrained to come from a predefined vocabulary, similar to our case. The paper discusses the idea of how to use traditional smoothing to generalize over different contexts and outcomes. In [9] the authors have addressed the problem of concept-to-text generation using discriminative ranking over hypergraphs that encode exponentially many derivations of a probabilistic CFG.

Moving on to deep learning techniques, in [7] Dusek et.al introduces the first fully trainable entertainment-enabled NLG system. It uses a deep learning based approach to response generation using RNN and sequence-to-sequence models with attention. The authors experiment on a dataset related to the public transport domain. The problem discussed is exactly similar to the one we are solving here, except that we are proposing an unsupervised approach. There have been several other studies on applying deep learning for NLG. In [19] the authors propose a RNN based approach to context-aware text generation by encoding the context into a continuous semantic representation and then decoding the representation to text sequences. [12] proposes a novel CNN based sequence-to-sequence model for NLG that allows building of hierarchical model which encapsulates word dependencies via shorter path than RNN. [22], like us, also tries to solve the problem of rule based QA systems by proposing a context-aware LSTM (CA-LSTM) model for NLG which is reported to have obtained state-of-the-art performance. However all such methods are supervised and are highly data driven.

Reinforcement Learning techniques present a natural choice for unsupervised approaches to many different tasks. [20] is a tutorial that summarizes techniques for applying Deep Reinforcement Learning (DRL) to NLG while [17] proposes an Inverse Reinforcement Learning (IRL) based solution to text-generation. Finally, in [10] the authors describe a constrained-NLG technique using Monte Carlo Tree Search (MCTS). The paper describes the similarity between the game of Go and NLG w.r.t the evaluation of the output being deferred to the very last step. In order to carry out the evaluation of the generated sentences, the paper proposes a syntactical score and an n-gram model based score. While the syntactical score ensures generated sentences are grammatically correct, the n-gram model score ensures that it makes sense. Though the results reported are preliminary, this technique can be explored further due its simplicity and ease of tuning to improve results.

4. Dataset

We are using a private data set from WalmartLabs. The data set consists of 20000 synthetic and live examples. The i^{th} data point d_i is a 4-tuple given by

$$d_i = (u_i, I_i, a_i, e_i)$$

where u_i is the user question string, I_i is the intent, a_i is the answer slot and e_i is any named entity if present. An example data row is {"*do you have milk*", "*search*", "*low fat milk priced at 3 dollars*", "*milk*"}

Test Set: 40 queries from this dataset, spanning 10 intents (4 paraphrases for each intent) have been hand picked by analysts as a small test group that adequately represents the entire gamut of conversations users normally have with the system. As explained in section 2, for each of these 40 examples, given u_i and a_i our task is to generate an entertaining response $\mathcal{R}$.

Conversational Data Set: Though our core solution is unsupervised, one of the solution components involve building a classifier that can classify valid vs invalid sentences. Since we are dealing with predominantly conversational data, we decided to use the Cornell Movie-Dialogues Corpus [5] that has 220579 conversational exchanges between 10292 pairs of movie characters.

Pre-processing: A salient feature of the movie dialogue data set is that it is heavily contracted, i.e. "I am", "there is" etc are always mentioned as "I'm", "tehre's" etc. We built a map from contracted to un-contracted words and passed the extracted set of dialogues through a regex filter to capture contracted forms and replace then with their expanded forms. We followed this by other normalization steps like converting everything to lower case, splitting multi sentence rows into individual lines and removing other punctuation and special characters like quotes and ellipsis. After the pre processing step, we got a total of **516281** sentences. Details of training, test and validation data set generation and feature extraction is described in the next section when we describe the sub-component that requires this data

5. Method

We begin with 2 observations

- When initiating a conversation in the e-commerce domain, user queries are paraphrased either as a command (e.g. "buy apples") or as a question (e.g. "can you get me some apples", with the question type being predominant.
- Entertainment (as used in our context) is only relevant when users paraphrase their queries as questions since the other type, commands, are linguistically neutral and can be answered in many different ways.

Assumption: For questions, entertaining responses can be generated by reusing words from the original user utterance after inducing some, possibly empty, transformation on them and permuting their order. This assumption is valid for a large number of questions.

Target ans: Let $\mathcal{P}(u)$ denote the set of all unique permutations of the ordered sequence u, with $\mathcal{P}_k(u)$ denoting one such permutation. Let $\mathcal{T}(u_i)$ denote a transformation operation on an element of the ordered sequence u where the transformation can be either a deletion or change of tense, person etc. Then above assumption, along with Eqn 1 and Eqn 2 imply that the expected response $\mathcal{R}$ can be written as

$$\mathcal{R} = \mathcal{P}(\mathcal{T}(u_1), \mathcal{T}(u_2), ...\mathcal{T}(u_n), a) \tag{3}$$

In other words, we are assuming that the target answer can be obtained by suitably the transformed words of the original user utterance and the un-transformed answer slot.

State and Transformation: We model the problem as a discrete state single-player game play problem, where the state of the game at point t is given by $\mathcal{S}_t$ the sentence formed so far. The state transition operators $\mathcal{P}$ and $\mathcal{T}$ operate on $\mathcal{S}$ to take it to a new game state. A game-state evaluation function $\mathcal{E}$, evaluates the game state at each step t and provides feedback to the agent. The agent stops after making a pre-defined number of max moves.

5.1. Evaluation function

We define the evaluation function as follows

$$\mathcal{E}(s) = t(s) * \{c * k(s) + (1 - c) * l(s) + b(s, u)\} \tag{4}$$

where $0 \leq c,\ t(s),\ k(s),\ l(s),\ b(s, u) \leq 1$. We describe each of these terms briefly.

Syntactical correctness: $t(s)$ is the probability of the sentence being syntactically correct. We extract 16000 sentences of length between 5 to 10 words from the cleaned Cornell Movie-Dialogue dataset described in section 4. These are our valid sentences. For each of these valid sentences we generate 3 invalid sentences by randomly permuting the valid sentences. This gives us 48000 invalid sentences. We then use the Stanford NLP Parser to obtain the dependency parsed trees for all 64000 of them. From each tree, we extract the following features

- All non-terminal nodes in in-order traversal
- All non-terminal nodes in pre-order traversal
- All sub-tree roots up to a depth of height - 1 and their first level children.

These feature sets, properly tagged as valid and invalid are divided into validation set (1000 each of valid and invalid feature sets), test set (1000 each of valid and invalid feature

sets) and training set (all the rest). Using this we train a Naive Baye's classifier which has an accuracy of 87% on the training set. The final output for $t(s)$ is overridden to be 0, irrespective of the classifier output if, the input sentence contains any of FRAG, SINV, SBARQ, WHADVP, or SQ nodes, since these always indicate a question being asked, which the generated sentence is trying to answer.

Similarity to the users utterance: $b(s, u)$ the smoothed BLEU-4 score of the generated sentence s when compared to the reference which is the user utterance u. High BLEU score is desirable.

Probability of s: $k(s)$ measures the expected Kneser-Ney smoothed trigram probability over the entire sentence, while $l(s)$ measures the expected Jelinek-Mercer smoothed bi-gram probabilities over the entire generated sentence. c is a fixed empirically determined constant that weights the contribution of $k(s)$ vs $l(s)$. c is set to 0.2

5.2. Baseline: Random Sampling

We first create a baseline random sampling solution to this problem. At each trial the algorithm chooses either the $\mathcal{P}$ or the $\mathcal{T}$ operator randomly and applies them on S. This continues for a fixed number of trials and we take the best sentence generated.

Data: u, a
Result: s^*: The best sentence generated
Initialize $s_0 = u$;
maxIter = 1000;
while *iter* < *maxIter* **do**
  O = randomChoice ($\mathcal{P}$,$\mathcal{T}$);
  s = $\mathcal{O}(s)$;
  s^* = max(s^*, s);
  iter += 1;
end

Algorithm 1: Baseline random sampling

5.3. Monte Carlo Tree Search

For a more effective approach we carry out the search over the state space using Monte Carlo Tree Search (MCTS). We use the same variant of the algorithm as described in [10], except we use our own evaluation function defined in Eqn 4. We use two different bandit algorithms to select nodes. The first one is the Upper Confidence Bound (UCB) given by

$$\text{UCB1} = v_i + C\sqrt{\frac{\log N}{n_i}} \tag{5}$$

where v_i is the avg score of the outcomes (sentences) that went through this node, $C = \sqrt{2}$, N is the total number of trials so far, and n_i is the total number of visits to the i^{th} candidate so far.

For the PUCT variant we use the the following to determine the transition, where symbols have the same meaning as described here [2]

$$\hat{V}(z, a) + \sqrt{\frac{n(z)^{e(d)}}{n(z, a)}} \tag{6}$$

As shown in figure 3, whenever the MCT makes a move from a state S_k to S_m we store the previous and next steps in a separate file, which we plan to use later as explained in the future work section.

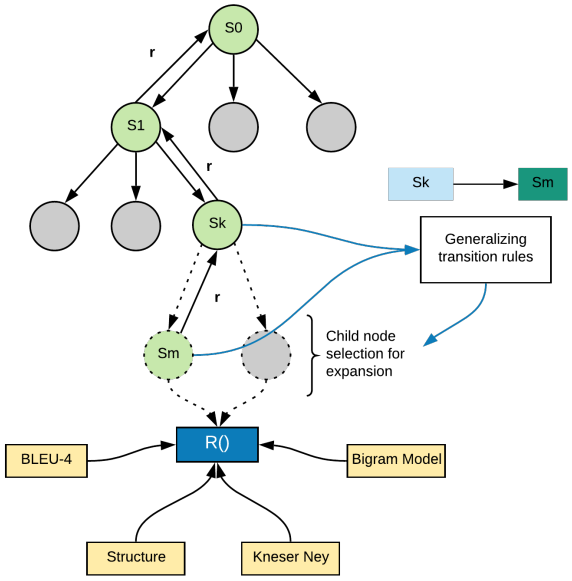

Figure 3. Execution flow of a conversational grocery assistant

6. Result

Here are some of our best results with the scores as measured by $\mathcal{E}$ mentioned beside them.

User: do you have any apples that are sold individually

Agent: i do have gala apples that are sold individually [**0.648**]

User: which apples are sold individually

Agent: gala apples are sold individually [**0.719**]

User: can you get me some apples that are sold individually

Agent: i can get you gala apples that are sold individually [**0.67**]

The following table describes our result in detail. The data set was so chosen so as to have 10 short queries (less than 5 words), 20 medium queries (between 5 to 10 words) and 10 long queries (more than 10 words). We ran the our 3 variants on all groups and observed the results. The algorithm performs well for short queries and performance deteriorates, both in terms of quality of sentences, and run time for longer queries. It is interesting to note that our empirically designed evaluation function $\mathcal{E}$ has good correlation with human evaluation, though we need to investigate this more rigorously.

Length	Bandit	Score	Human
$<= 5$	UCT	0.7129	80.0%
	PUCT	0.6990	70.0%
	Random	0.4127	50.0%
$5 - 10$	UCT	0.6274	55.0%
	PUCT	0.6035	60.0%
	Random	0.3255	10.0%
$>= 11$	UCT	0.5535	40.0%
	PUCT	0.5932	40.0%
	Random	0.3881	0.0%

7. Discussion

- Though some results were encouraging, the reward function needs more tuning to correlate more strongly to human evaluation.
- There is no standard implementation of Kneser-Ney smoothing for trigrams. Current NLTK implementation is buggy.
- Though it is normally ill-advised to use BLEU score at sentence level, we managed to use it to our advantage with appropriate smoothing and defensive coding.
- Naive Baye's worked surprisingly well for valid vs invalid sentence classification, when the invalid sentences were generated by sampling randomly from the corpus.

8. Future Work

This is an area of active research at WalmartLabs and efforts to improve and develop this approach will continue beyond the end of this course. Following will be the main areas of focus

- In addition to deterministic set of transition rules and unsupervised NLG, we will also explore if we can use a stochastic action space and learn the best moves in a supervised setting.
- The reward function can be parameterized, with different weights for each of the component objectives, which can then be learned from a held out data set.
- We will explore Expert Iteration System that generalizes the learning from the state transitions, which can then be used as a feedback to MCTS at the simulation step.

9. Project Code

The project code base is available at github: https://github.com/isnehasish/cs229-proj-code

10. Contribution

This project was completed by mukherji@stanford.edu working alone.

References

[1] L. E. Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman, "Frames: a corpus for adding memory to goal-oriented dialogue systems," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, 2017, pp. 207–219. [Online]. Available: https://aclanthology.info/papers/W17-5526/w17-5526

[2] D. Auger, A. Couëtoux, and O. Teytaud, "Continuous upper confidence trees with polynomial exploration - consistency," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I*, 2013, pp. 194–209. [Online]. Available: https://doi.org/10.1007/978-3-642-40988-2_13

[3] P. Budzianowski, T. Wen, B. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gasic, "Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling," *CoRR*, vol. abs/1810.00278, 2018. [Online]. Available: http://arxiv.org/abs/1810.00278

[4] S. Chen, D. Beeferman, and R. Rosenfeld, "Evaluation metrics for language models," 1998.

[5] C. Danescu-Niculescu-Mizil and L. Lee, "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs." in *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

[6] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston, "Wizard of wikipedia: Knowledge-powered conversational agents," *CoRR*, vol. abs/1811.01241, 2018. [Online]. Available: http://arxiv.org/abs/1811.01241

[7] O. Dusek and F. Jurcícek, "A context-aware natural language generator for dialogue systems," in *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, 2016, pp. 185–190. [Online]. Available: http://aclweb.org/anthology/W/W16/W16-3622.pdf

[8] M. Henderson, B. Thomson, and J. W. and, "The second dialog state tracking challenge. in proceedings of sigdial," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, 2014, pp. 207–219. [Online]. Available: https://www.aclweb.org/anthology/W14-4337

[9] I. Konstas and M. Lapata, "Concept-to-text generation via discriminative reranking," in *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, 2012, pp. 369–378. [Online]. Available: http://www.aclweb.org/anthology/P12-1039

[10] K. Kumagai, I. Kobayashi, D. Mochihashi, H. Asoh, T. Nakamura, and T. Nagai, "Human-like natural language generation using monte carlo tree search," in *Proceedings of the INLG 2016 Workshop on Computational Creativity in Natural Language Generation, CC-NLG 2016, Edinburgh, UK, September 2016*, 2016, pp. 11–18. [Online]. Available: https://doi.org/10.18653/v1/W16-5502

[11] O. Lemon, "Adaptive natural language generation in dialogue using reinforcement learning," in *Proceedings of the 12th Workshop on the Semantics and Pragmatics of Dialogue (LONDIAL)*, ser. Proceedings (SemDial), 2008, pp. 141–148.

[12] S. Mangrulkar, S. Shrivastava, V. Thenkanidiyoor, and D. A. Dinesh, "A context-aware convolutional natural language generation model for dialogue systems," in *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, July 12-14, 2018*, 2018, pp. 191–200. [Online]. Available: https://aclanthology.info/papers/W18-5020/w18-5020

[13] P. Mazaré, S. Humeau, M. Raison, and A. Bordes, "Training millions of personalized dialogue agents," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2018, pp. 2775–2779. [Online]. Available: https://aclanthology.info/papers/D18-1298/d18-1298

[14] P. Shah, D. Hakkani-Tür, G. Tür, A. Rastogi, A. Bapna, N. Nayak, and L. P. Heck, "Building a conversational agent overnight with dialogue self-play," *CoRR*, vol. abs/1801.04871, 2018. [Online]. Available: http://arxiv.org/abs/1801.04871

[15] Y. Shao, S. Gouws, D. Britz, A. Goldie, B. Strope, and R. Kurzweil, "Generating high-quality and informative conversation responses with sequence-to-sequence models," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2017, pp. 2210–2219. [Online]. Available: https://aclanthology.info/papers/D17-1235/d17-1235

[16] Y. Shen, J. Chen, P. Huang, Y. Guo, and J. Gao, "M-walk: Learning to walk over graphs using monte carlo tree search," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, 2018, pp. 6787–6798. [Online]. Available: http://papers.nips.cc/paper/7912-m-walk-learning-to-walk-over-graphs-using-monte-carlo-tree-s

[17] Z. Shi, X. Chen, X. Qiu, and X. Huang, "Towards diverse text generation with inverse reinforcement learning," *CoRR*, vol. abs/1804.11258, 2018. [Online]. Available: http://arxiv.org/abs/1804.11258

[18] B. Swanson, E. Yamangil, and E. Charniak, "Natural language generation with vocabulary constraints," in *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2014, June 26, 2014, Baltimore, Maryland, USA*, 2014, pp. 124–133. [Online]. Available: http://aclweb.org/anthology/W/W14/W14-1815.pdf

[19] J. Tang, Y. Yang, S. Carton, M. Zhang, and Q. Mei, "Context-aware natural language generation with recurrent neural networks," *CoRR*, vol. abs/1611.09900, 2016. [Online]. Available: http://arxiv.org/abs/1611.09900

[20] W. Y. Wang, J. Li, and X. He, "Deep reinforcement learning for NLP," in *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, Tutorial Abstracts*, 2018, pp. 19–21. [Online]. Available: https://aclanthology.info/papers/P18-5007/p18-5007

[21] C. Xiao, J. Mei, and M. Müller, "Memory-augmented monte carlo tree search," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI*

Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, 2018, pp. 1455–1462. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17139

[22] H. Zhou, M. Huang, and X. Zhu, "Context-aware natural language generation for spoken dialogue systems," in *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, 2016, pp. 2032–2041. [Online]. Available: http://aclweb.org/anthology/C/C16/C16-1191.pdf