# Stochastic Language Generation in Dialogue using Recurrent Neural Networks with Convolutional Sentence Reranking

**Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić,**
**Pei-Hao Su, David Vandyke** and **Steve Young**
Cambridge University Engineering Department,
Trumpington Street, Cambridge, CB2 1PZ, UK
{thw28,mg436,dk449,nm480,phs26,djv27,sjy}@cam.ac.uk

## Abstract

The natural language generation (NLG) component of a spoken dialogue system (SDS) usually needs a substantial amount of handcrafting or a well-labeled dataset to be trained on. These limitations add significantly to development costs and make cross-domain, multi-lingual dialogue systems intractable. Moreover, human languages are context-aware. The most natural response should be directly learned from data rather than depending on predefined syntaxes or rules. This paper presents a statistical language generator based on a joint recurrent and convolutional neural network structure which can be trained on dialogue act-utterance pairs without any semantic alignments or predefined grammar trees. Objective metrics suggest that this new model outperforms previous methods under the same experimental conditions. Results of an evaluation by human judges indicate that it produces not only high quality but linguistically varied utterances which are preferred compared to n-gram and rule-based systems.

## 1 Introduction

Conventional spoken dialogue systems (SDS) are expensive to build because many of the processing components require a substantial amount of handcrafting (Ward and Issar, 1994; Bohus and Rudnicky, 2009). In the past decade, significant progress has been made in applying statistical methods to automate the speech understanding and dialogue management components of an SDS, including making them more easily extensible to other application domains (Young et al., 2013; Gašić et al., 2014; Henderson et al., 2014). However, due to the difficulty of collecting semantically-annotated corpora, the use of data-driven NLG for SDS remains relatively unexplored and rule-based generation remains the norm for most systems (Cheyer and Guzzoni, 2007; Mirkovic and Cavedon, 2011).

The goal of the NLG component of an SDS is to map an abstract dialogue act consisting of an act type and a set of attribute-value pairs[1] into an appropriate surface text (see Table 1 below for some examples). An early example of a statistical NLG system is HALOGEN by Langkilde and Knight (1998) which uses an n-gram language model (LM) to rerank a set of candidates generated by a handcrafted generator. In order to reduce the amount of handcrafting and make the approach more useful in SDS, Oh and Rudnicky (2000) replaced the handcrafted generator with a set of word-based n-gram LM-based generators, one for each dialogue type and then reranked the generator outputs using a set of rules to produce the final response. Although Oh and Rudnicky (2000)'s approach limits the amount of handcrafting to a small set of post-processing rules, their system incurs a large computational cost in the over-generation phase and it is difficult to ensure that all of the required semantics are covered by the selected output. More recently, a phrase-based NLG system called BAGEL trained from utterances aligned with coarse-grained semantic concepts has been described (Mairesse et al., 2010; Mairesse and Young, 2014). By implicitly modelling paraphrases, Bagel can generate linguistically varied utterances. However, collecting semantically-aligned corpora is expensive and time consuming, which limits Bagel's scalability to new domains.

This paper presents a neural network based NLG system that can be fully trained from dia-

[1] Here and elsewhere, attributes are frequently referred to as *slots*.



*Proceedings of the SIGDIAL 2015 Conference*, pages 275–284,
Prague, Czech Republic, 2-4 September 2015. 

log act-utterance pairs without any semantic alignments between the two. We start in Section 3 by presenting a generator based on a recurrent neural network language model (RNNLM) (Mikolov et al., 2010; Mikolov et al., 2011a) which is trained on a *delexicalised* corpus (Henderson et al., 2014) whereby each value has been replaced by a symbol representing its corresponding slot. In a final post-processing phase, these slot symbols are converted back to the corresponding slot values.

While generating, the RNN generator is conditioned on an auxiliary dialogue act feature and a controlling gate to over-generate candidate utterances for subsequent reranking. In order to account for arbitrary slot-value pairs that cannot be routinely delexicalized in our corpus, Section 3.1 describes a convolutional neural network (CNN) (Collobert and Weston, 2008; Kalchbrenner et al., 2014) sentence model which is used to validate the semantic consistency of candidate utterances during reranking. Finally, by adding a backward RNNLM reranker into the model in Section 3.2, output fluency is further improved. Training and decoding details of the proposed system are described in Section 3.3 and 3.4.

Section 4 presents an evaluation of the proposed system in the context of an application providing information about restaurants in the San Francisco area. In Section 4.2, we first show that new generator outperforms Oh and Rudnicky (2000)'s utterance class LM approach using objective metrics, whilst at the same time being more computationally efficient. In order to assess the subjective performance of our system, pairwise preference tests are presented in Section 4.3. The results show that our approach can produce high quality utterances that are considered to be more natural than a rule-based generator. Moreover, by sampling utterances from the top reranked output, our system can also generate linguistically varied utterances. Section 4.4 provides a more detailed analysis of the contribution of each component of the system to the final performance. We conclude with a brief summary and future work in Section 5.

## 2 Related Work

Conventional approaches to NLG typically divide the task into sentence planning, and surface realisation. Sentence planning maps input semantic symbols into an intermediary tree-like or template structure representing the utterance, then surface realisation converts the intermediate structure into the final text (Walker et al., 2002; Stent et al., 2004; Dethlefs et al., 2013). As noted above, one of the first statistical NLG methods that requires almost no handcrafting or semantic alignments was an n-gram based approach by Oh and Rudnicky (2000). Ratnaparkhi (2002) later addressed the limitations of n-gram LMs in the over-generation phase by using a more sophisticated generator based on a syntactic dependency tree.

Statistical approaches have also been studied for sentence planning, for example, generating the most likely context-free derivations given a corpus (Belz, 2008) or maximising the expected reward using reinforcement learning (Rieser and Lemon, 2010). Angeli et al. (2010) train a set of log-linear models to predict individual generation decisions given the previous ones, using only domain-independent features. Along similar lines, by casting NLG as a template extraction and reranking problem, Kondadadi et al. (2013) show that outputs produced by an SVM reranker are comparable to human-authored texts.

The use of neural network-based approaches to NLG is relatively unexplored. The stock reporter system ANA by Kukich (1987) is a network based NLG system, in which the generation task is divided into a sememe-to-morpheme network followed by a morpheme-to-phrase network. Recent advances in recurrent neural network-based language models (RNNLM) (Mikolov et al., 2010; Mikolov et al., 2011a) have demonstrated the value of distributed representations and the ability to model arbitrarily long dependencies for both speech recognition and machine translation tasks. Sutskever et al. (2011) describes a simple variant of the RNN that can generate meaningful sentences by learning from a character-level corpus. More recently, Karpathy and Fei-Fei (2014) have demonstrated that an RNNLM is capable of generating image descriptions by conditioning the network model on a pre-trained convolutional image feature representation. This work provides a key inspiration for the system described here. Zhang and Lapata (2014) describes interesting work using RNNs to generate Chinese poetry.

A specific requirement of NLG for dialogue systems is that the concepts encoded in the abstract system dialogue act must be conveyed accurately by the generated surface utterance, and simple unconstrained RNNLMs which rely on em-

bedding at the word level (Mikolov et al., 2013; Pennington et al., 2014) are rather poor at this. As a consequence, new methods have been investigated to learn distributed representations for phrases and even sentences by training models using different structures (Collobert and Weston, 2008; Socher et al., 2013). Convolutional Neural Networks (CNNs) were first studied in computer vision for object recognition (Lecun et al., 1998). By stacking several convolutional-pooling layers followed by a fully connected feed-forward network, CNNs are claimed to be able to extract several levels of translational-invariant features that are useful in classification tasks. The convolutional sentence model (Kalchbrenner et al., 2014; Kim, 2014) adopts the same methodology but collapses the two dimensional convolution and pooling process into a single dimension. The resulting model is claimed to represent the state-of-the-art for many speech and NLP related tasks (Kalchbrenner et al., 2014; Sainath et al., 2013).

## 3 Recurrent Generation Model

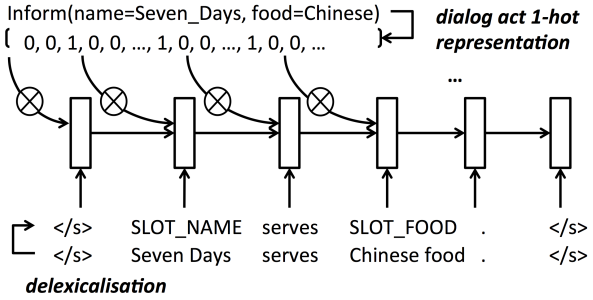


Figure 1: An unrolled view of the RNN-based generation model. It operates on a delexicalised utterance and a 1-hot encoded feature vector specified by a dialogue act type and a set of slot-value pairs. $\otimes$ indicates the gate used for controlling the on/off states of certain feature values. The output connection layer is omitted here for simplicity.

The generation model proposed in this paper is based on an RNNLM architecture (Mikolov et al., 2010) in which a 1-hot encoding $\mathbf{w}_t$ of a token[2] $w_t$ is input at each time step $t$ conditioned on a recurrent hidden layer $\mathbf{h}_t$ and outputs the probability distribution of the next token $w_{t+1}$. Therefore, by sampling input tokens one by one from the output distribution of the RNN until a stop sign is generated (Karpathy and Fei-Fei, 2014) or some required constraint is satisfied (Zhang and Lapata, 2014), the network can produce a sequence of tokens which can be lexicalised to form the required utterance.

In order to ensure that the generated utterance represents the intended meaning, the input vectors $\mathbf{w}_t$ are augmented by a control vector $\mathbf{f}$ constructed from the concatenation of 1-hot encodings of the required dialogue act and its associated slot-value pairs. The auxiliary information provided by this control vector tends to decay over time because of the *vanishing gradient problem* (Mikolov and Zweig, 2012; Bengio et al., 1994). Hence, $\mathbf{f}$ is reapplied to the RNN at every time step as in Karpathy and Fei-Fei (2014).

In detail, the recurrent generator shown in Figure 1 is defined as follows:

$$\mathbf{h}_t = sigmoid(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{wh}\mathbf{w}_t + \mathbf{W}_{fh}\mathbf{f}_t) \quad (1)$$

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

$$w_{t+1} \sim P(w_{t+1}|w_t, w_{t-1}, ...w_0, \mathbf{f}_t) \quad (3)$$

where $\mathbf{W}_{hh}$, $\mathbf{W}_{wh}$, $\mathbf{W}_{fh}$, and $\mathbf{W}_{ho}$ are the learned network weight matrices. $\mathbf{f}_t$ is a gated version of $\mathbf{f}$ designed to discourage duplication of information in the generated output in which each segment $\mathbf{f}_s$ of the control vector $\mathbf{f}$ corresponding to slot $s$ is replaced by

$$\mathbf{f}_{s,t} = \mathbf{f}_s \odot \delta^{t-t_s} \quad (4)$$

where $t_s$ is the time at which slot $s$ first appears in the output, $\delta \leq 1$ is a decay factor, and $\odot$ denotes element-wise multiplication. The effect of this gating is to decrease the probability of regenerating slot symbols that have already been generated, and to increase the probability of rendering all of the information encoded in $\mathbf{f}$.

The tokenisation resulting from delexicalising slots and values does not work for all cases. For example, some slot-value pairs such as *food=dont_care* or *kids_allowed=false* cannot be directly modelled using this technique because there is no explicit value to delexicalise in the training corpus. As a consequence, the model is prone to errors when these slot-value pairs are required. A further problem is that the RNNLM generator selects words based only on the preceding history, whereas some sentence forms depend on the backward context.

[2]We use *token* instead of *word* because our model operates on text for which slot names and values have been delexicalised.

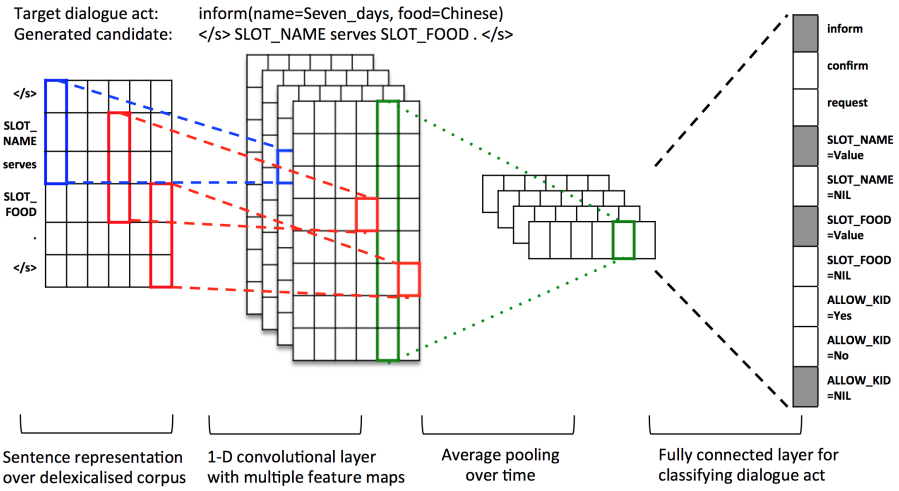


Figure 2: Our simple variant of CNN sentence model as described in Kalchbrenner et al. (2014).

To deal with these issues, candidates generated by the RNNLM are reranked using two models. Firstly, a convolutional neural network (CNN) sentence model (Kalchbrenner et al., 2014; Kim, 2014) is used to ensure that the required dialogue act and slot-value pairs are represented in the generated utterance, including the non-standard cases. Secondly, a *backward* RNNLM is used to rerank utterances presented in reverse order.

### 3.1 Convolutional Sentence Model

The CNN sentence model is shown in Figure 2. Given a candidate utterance of length $n$, an utterance matrix $\mathbf{U}$ is constructed by stacking embeddings $\mathrm{w_t}$ of each token in the utterance:

$$\mathbf{U} = \begin{bmatrix} \text{——} & \mathrm{w}_0 & \text{——} \\ \text{——} & \mathrm{w}_1 & \text{——} \\ & \cdots & \\ \text{——} & \mathrm{w}_{n-1} & \text{——} \end{bmatrix}. \quad (5)$$

A set of $K$ convolutional mappings are then applied to the utterance to form a set of feature detectors. The outputs of these detectors are combined and fed into a fully-connected feed-forward network to classify the action type and whether each required slot is mentioned or not.

Each mapping $k$ consists of a one-dimensional convolution between a filter $\mathbf{m}_\mathrm{k} \in \mathbb{R}^m$ and the utterance matrix $\mathbf{U}$ to produce another matrix $\mathbf{C}^\mathrm{k}$:

$$\mathbf{C}^\mathrm{k}{}_{i,j} = \mathbf{m}_\mathrm{k}{}^\intercal \mathbf{U}_{i-m+1:i,j} \quad (6)$$

where $m$ is the filter size, and $i$,$j$ is the row and column index respectively. The outputs of each column of $\mathbf{C}^\mathrm{k}$ are then pooled by averaging[3] over time:

$$\mathbf{h}_\mathrm{k} = \left[\bar{\mathbf{C}}^k_{:,0}, \bar{\mathbf{C}}^k_{:,1}, ..., \bar{\mathbf{C}}^k_{:,h-1}\right] \quad (7)$$

where $h$ is the size of embedding and $k = 1 \ldots K$. Last, the $K$ pooled feature vectors $\mathbf{h}_k$ are passed through a nonlinearity function to obtain the final feature map.

### 3.2 Backward RNN reranking

As noted earlier, the quality of an RNN language model may be improved if both forward and backward contexts are considered. Previously, bidirectional RNNs (Schuster and Paliwal, 1997) have been shown to be effective for handwriting recognition (Graves et al., 2008), speech recognition (Graves et al., 2013), and machine translation (Sundermeyer et al., 2014). However, applying a bidirectional RNN directly in our generator is not straightforward since the generation process is sequential in time. Hence instead of integrating the bidirectional information into a single unified network, the forward and backward contexts are utilised separately by firstly generating candidates using the forward RNN generator, then using the log-likelihood computed by a backward RNNLM to rerank the candidates.

### 3.3 Training

Overall the proposed generation architecture requires three models to be trained: a forward RNN generator, a CNN reranker, and a backward RNN reranker. The objective functions for training the

[3]Max pooling was also tested but was found to be inferior to average pooling

two RNN models are the cross entropy errors between the predicted word distribution and the actual word distribution in the training corpus, whilst the objective for the CNN model is the cross entropy error between the predicted dialogue act and the actual dialogue act, summed over the act type and each slot. An $l_2$ regularisation term is added to the objective function for every 10 training examples as suggested in Mikolov et al. (2011b). The three networks share the same set of word embeddings, initialised with pre-trained word vectors provided by Pennington et al. (2014). All costs and gradients are computed and stochastic gradient descent is used to optimise the parameters. Both RNNs were trained with back propagation through time (Werbos, 1990). In order to prevent overfitting, early stopping was implemented using a held-out validation set.

### 3.4 Decoding

The decoding procedure is split into two phases: (a) over-generation, and (b) reranking. In the over-generation phase, the forward RNN generator conditioned on the given dialogue act, is used to sequentially generate utterances by random sampling of the predicted next word distributions. In the reranking phase, the hamming loss $cost_{CNN}$ of each candidate is computed using the CNN sentence model and the log-likelihood $cost_{bRNN}$ is computed using the backward RNN. Together with the log-likelihood $cost_{fRNN}$ from the forward RNN, the reranking score $R$ is computed as:

$$R = -(cost_{fRNN} + cost_{bRNN} + cost_{CNN}). \quad (8)$$

This is the reranking criterion used to analyse each individual model in Section 4.4.

Generation quality can be further improved by introducing a slot error criterion ERR, which is the *number of slots generated that is either redundant or missing*. This is also used in Oh and Rudnicky (2000). Adding this to equation (8) yields the final reranking score $R^*$:

$$R^* = -(cost_{fRNN} + cost_{bRNN} + cost_{CNN} + \lambda \mathrm{ERR}) \quad (9)$$

In order to severely penalise nonsensical utterances, $\lambda$ is set to 100 for both the proposed RNN system and our implementation of Oh and Rudnicky (2000)'s n-gram based system. This reranking criterion is used for both the automatic evaluation in Section 4.2 and the human evaluation in Section 4.3.

## 4 Experiments

### 4.1 Experimental Setup

The target application area for our generation system is a spoken dialogue system providing information about restaurants in San Francisco. There are 8 system dialogue act types such as *inform* to present information about restaurants, *confirm* to check that a slot value has been recognised correctly, and *reject* to advise that the user's constraints cannot be met (Table 1 gives the full list with examples); and there are 12 attributes (slots): *name, count, food, near, price, pricerange, postcode, phone, address, area, goodformeal, and kidsallowed*, in which all slots are categorical except *kidsallowed* which is binary.

To form a training corpus, dialogues from a set of 3577 dialogues collected in a user trial of a statistical dialogue manager proposed by Young et al. (2013) were randomly sampled and shown to workers recruited via the Amazon Mechanical Turk service. Workers were shown each dialogue turn by turn and asked to enter an appropriate system response in natural English corresponding to each system dialogue act. The resulting corpus contains 5193 hand-crafted system utterances from 1006 randomly sampled dialogues. Each categorical value was replaced by a token representing its slot, and slots that appeared multiple times in a dialogue act were merged into one. This resulted in 228 distinct dialogue acts.

The system was implemented using the Theano library (Bergstra et al., 2010; Bastien et al., 2012). The system was trained by partitioning the 5193 utterances into a training set, validation set, and testing set in the ratio 3:1:1, respectively. The frequency of each action type and slot-value pair differs quite markedly across the corpus, hence up-sampling was used to make the corpus more uniform. Since our generator works stochastically and the trained networks can differ depending on the initialisation, all the results shown below[4] were averaged over 10 randomly initialised networks. The BLEU-4 metric was used for the objective evaluation (Papineni et al., 2002). Multiple references for each test dialogue act were obtained by mapping them back to the 228 distinct dialogue acts, merging those delexicalised templates that have the same dialogue act specification, and then lexicalising those templates back to

[4]Except human evaluation, in which only one set of network was used.

Table 1: The 8 system dialogue acts with example realisations

| # | Dialogue act and example realisations of our system, by sampling from top-5 candidates |
|---|---|
| 1 | inform(name="stroganoff restaurant",pricerange=cheap,near="fishermans wharf")<br>*stroganoff restaurant is a cheap restaurant near fishermans wharf .*<br>*stroganoff restaurant is in the cheap price range near fishermans wharf .* |
| 2 | reject(kidsallowed=yes,food="basque")<br>*unfortunately there are 0 restaurants that allow kids and serve basque .* |
| 3 | informonly(name="bund shanghai restaurant", food="shanghainese")<br>*i apologize , no other restaurant except bund shanghai restaurant that serves shanghainese .*<br>*sorry but there is no place other than the restaurant bund shanghai restaurant for shanghainese .* |
| 4 | confirm(goodformeal=dontcare)<br>*i am sorry . just to confirm . you are looking for a restaurant good for any meal ?*<br>*can i confirm that you do not care about what meal they offer ?* |
| 5 | request(near)<br>*would you like to dine near a particular location ?* |
| 6 | reqmore()<br>*is there anything else i can do for you ?* |
| 7 | select(kidsallowed=yes, kidsallowed=no)<br>*are you looking for a restaurant that allows kids , or does not allow kids ?* |
| 8 | goodbye()<br>*thank you for calling . good bye .* |

Table 2: Comparison of top-1 utterance between the RNN-based system and three baselines. A two-tailed Wilcoxon rank sum test was applied to compare the RNN model with the best O&R system (the 3-slot, 5g configuration) over 10 random seeds. (*=p<.005)

| Method | beam | BLEU | ERR |
|---|---|---|---|
| handcrafted | n/a | 0.440 | 0 |
| kNN | n/a | 0.591 | 17.2 |
| O&R,0-slot,5g | 1/20 | 0.527 | 635.2 |
| O&R,1-slot,5g | 1/20 | 0.610 | 460.8 |
| O&R,2-slot,5g | 1/20 | 0.719 | 142.0 |
| O&R,3-slot,3g | 1/20 | 0.760 | 74.4 |
| O&R,3-slot,4g | 1/20 | 0.758 | 53.2 |
| O&R,3-slot,5g | 1/20 | 0.757 | 47.8 |
| Our Model | 1/20 | **0.777*** | **0*** |

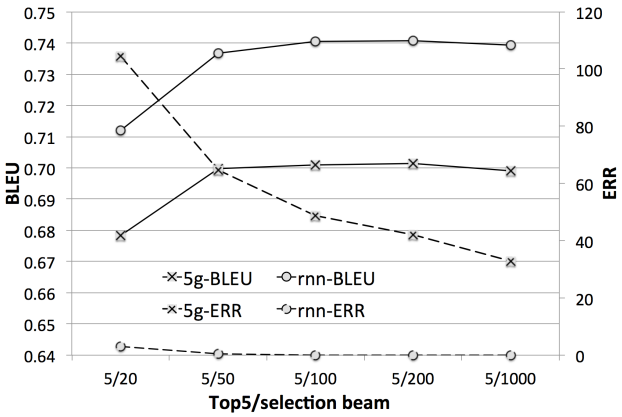


Figure 3: Comparison of our method (rnn) with O&R's approach (5g) in terms of optimising top-5 results over different selection beams.

form utterances. In addition, the slot error (ERR) as described in Section 3.4, out of 1848 slots in 1039 testing examples, was computed alongside the BLEU score.

### 4.2 Empirical Comparison

As can be seen in Table 2, we compare our proposed RNN-based method with three baselines: a handcrafted generator, a k-nearest neighbour method (kNN), and Oh and Rudnicky (2000)'s n-gram based approach (O&R). The handcrafted generator was tuned over a long period of time and has been used frequently to interact with real users. We found its performance is reliable and robust. The kNN was performed by computing the similarity of the testing dialogue act 1-hot vector against all training examples. The most similar template in the training set was then selected and lexicalised as the testing realisation. We found our RNN generator significantly outperforms these two approaches. While comparing with the O&R system, we found that by partitioning the corpus into more and more utterance classes, the O&R system can also reach a BLEU score of 0.76. However, the slot error cannot be efficiently reduced to zero even when using the error itself as a reranking criterion. This problem is also noted in Mairesse and Young (2014). In contrast, the RNN system produces utterances without slot errors when reranking using the same number of candidates, and it achieves the highest BLEU score. Figure 3 compares the RNN system with O&R's system when randomly select-

Table 3: Pairwise comparison between four systems. Two quality evaluations (rating out of 5) and one preference test were performed in each case. Statistical significance was computed using a two-tailed Wilcoxon rank sum test and a two-tailed binomial test (*=p<.05, **=p<.005).

| Metrics | handcrafted | $RNN_1$ | handcrafted | $RNN_5$ | $RNN_1$ | $RNN_5$ | $O\&R_5$ | $RNN_5$ |
|---|---|---|---|---|---|---|---|---|
| | 148 dialogs, 829 utt. | | 148 dialogs, 814 utt. | | 144 dialogs, 799 utt. | | 145 dialogs, 841 utt. | |
| Info. | 3.75 | 3.81 | 3.85 | 3.93* | 3.75 | 3.72 | 4.02 | 4.15* |
| Nat. | 3.58 | 3.74** | 3.57 | 3.94** | 3.67 | 3.58 | 3.91 | 4.02 |
| Pref. | 44.8% | 55.2%* | 37.2% | 62.8%** | 47.5% | 52.5% | 47.1% | 52.9% |

ing from the top-5 ranked results in order to introduce linguistic diversity. Results suggest that although O&R's approach improves as the selection beam increases, the RNN-based system is still better in both metrics. Furthermore, the slot error of the RNN system drops to zero when the selection beam is around 50. This indicates that the RNN system is capable of generating paraphrases by simply increasing the number of candidates during the over-generation phase.

### 4.3 Human Evaluation

Whilst automated metrics provide useful information for comparing different systems, human testing is needed to assess subjective quality. To do this, about 60 judges were recruited using Amazon Mechanical Turk and system responses were generated for the remaining 2571 unseen dialogues mentioned in Section 4.1. Each judge was then shown a randomly selected dialogue, turn by turn. At each turn, two utterances were generated from two different systems and presented to the judge who was asked to score each utterance in terms of informativeness and naturalness (rating out of 5), and also asked to state a preference between the two taking account of the given dialogue act and the dialogue context. Here *informativeness* is defined as whether the utterance contains all the information specified in the dialogue act, and *naturalness* is defined as whether the utterance could have been produced by a human. The trial was run pairwise across four systems: the RNN system using 1-best utterance $RNN_1$, the RNN system sampling from the top 5 utterances $RNN_5$, the O&R approach sampling from top 5 utterances $O\&R_5$, and a handcrafted baseline.

The result is shown in Table 3. As can be seen, the human judges preferred both $RNN_1$ and $RNN_5$ compared to the rule-based generator and the preference is statistically significant. Furthermore, the RNN systems scored higher in both informativeness and naturalness metrics, though the difference for informativeness is not statistically significant. When comparing $RNN_1$ with $RNN_5$, $RNN_1$ was judged to produce higher quality utterances but overall the diversity of output offered by $RNN_5$ made it the preferred system. Even though the preference is not statistically significant, it echoes previous findings (Pon-Barry et al., 2006; Mairesse and Young, 2014) that showed that language variability by paraphrasing in dialogue systems is generally beneficial. Lastly, $RNN_5$ was thought to be significantly better than O&R in terms of informativeness. This result verified our findings in Section 4.2 that O&R suffers from high slot error rates compared to the RNN system.

### 4.4 Analysis

In order to better understand the relative contribution of each component in the RNN-based generation process, a system was built in stages training first only the forward RNN generator, then adding the CNN reranker, and finally the whole model including the backward RNN reranker. Utterance candidates were reranked using Equation (8) rather than (9) to minimise manual intervention. As previously, the BLEU score and slot error (ERR) were measured.

**Gate** The forward RNN generator was trained first with different feature gating factors $\delta$. Using a selection beam of 20 and selecting the top 5 utterances, the result is shown in Figure 4 for $\delta$=1 is (equivalent to not using the gate), $\delta$=0.7, and $\delta$=0 (equivalent to turning off the feature immediately its corresponding slot has been generated). As can be seen, use of the feature gating substantially improves both BLEU score and slot error, and the best performance is achieved by setting $\delta$=0.

**CNN** The feature-gated forward RNN generator was then extended by adding a single convolutional-pooling layer CNN reranker. As shown in Figure 5, evaluation was performed on both the original dataset (*all*) and the dataset containing only binary slots and don't care values (*hard*). We found that the CNN reranker can better handle slots and values that cannot be explicitly

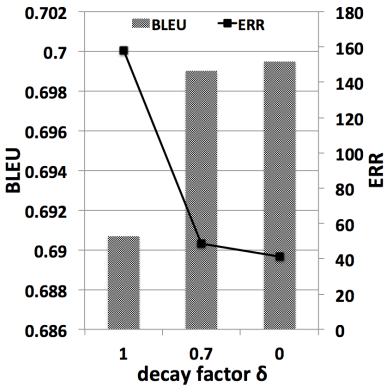


Figure 4: Feature gating effect

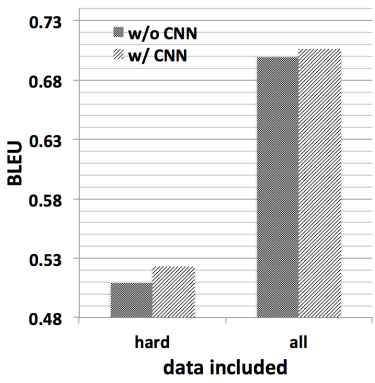


Figure 5: CNN effect

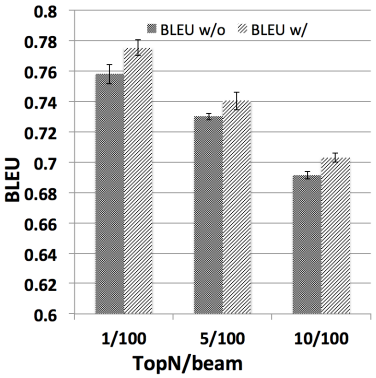


Figure 6: Backward RNN effect

delexicalised (1.5% improvement on *hard* comparing to 1% less on *all*).

**Backward RNN** Lastly, the backward RNN reranker was added and trained to give the full generation model. The selection beam was fixed at 100 and the $n$-best top results from which to select the output utterance was varied as $n = 1$, 5 and 10, trading accuracy for linguistic diversity. In each case, the BLEU score was computed with and without the backward RNN reranker. The results shown in Figure 6 are consistent with Section 4.2, in which BLEU score degraded as more $n$-best utterances were chosen. As can be seen, the backward RNN reranker provides a stable improvement no matter which value $n$ is.

**Training corpus size** Finally, Figure 7 shows the effect of varying the size of the training corpus. As can be seen, if only the 1-best utterance is offered to the user, then around 50% of the data (2000 utterances) is sufficient. However, if the linguistic variability provided by sampling from the top-5 utterances is required, then the figure suggest that more than 4156 utterances in the current training set are required.

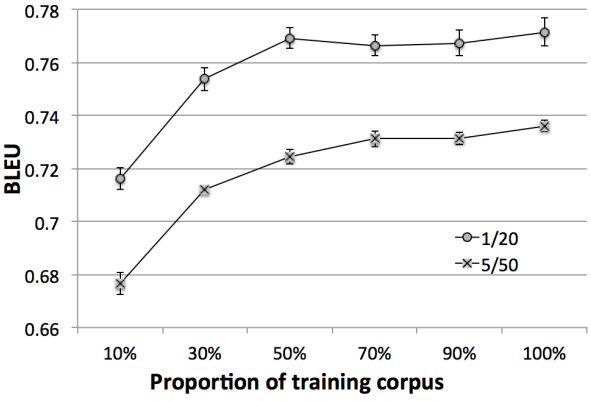


Figure 7: Networks trained with different proportion of data evaluated on two selection schemes.

## 5 Conclusion and Future Work

In this paper a neural network-based natural language generator has been presented in which a forward RNN generator, a CNN reranker, and backward RNN reranker are jointly optimised to generate utterances conditioned by the required dialogue act. The model can be trained on any corpus of dialogue act-utterance pairs without any semantic alignment and heavy feature engineering or handcrafting. The RNN-based generator is compared with an $n$-gram based generator which uses similar information. The $n$-gram generator can achieve similar BLEU scores but it is less efficient and prone to making errors in rendering all of the information contained in the input dialogue act.

An evaluation by human judges indicated that our system can produce not only high quality but linguistically varied utterances. The latter is particularly important in spoken dialogue systems where frequent repetition of identical output forms can rapidly become tedious.

The work reported in this paper is part of a larger programme to develop techniques for implementing open domain spoken dialogue. A key potential advantage of neural network based language processing is the implicit use of distributed representations for words and a single compact parameter encoding of a wide range of syntactic/semantic forms. This suggests that it should be possible to transfer a well-trained generator of the form proposed here to a new domain using a much smaller set of adaptation data. This will be the focus of our future work in this area.

## 6 Acknowledgements

Tsung-Hsien Wen and David Vandyke are supported by Toshiba Research Europe Ltd, Cambridge Research Laboratory.

## References


Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 502–512. Association for Computational Linguistics.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455, October.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference*.

Dan Bohus and Alexander I. Rudnicky. 2009. The ravenclaw dialog management framework: Architecture and systems. *Computer Speech and Language*, 23(3):332–361, July.

Adam Cheyer and Didier Guzzoni. 2007. Method and apparatus for building an intelligent automated assistant. US Patent App. 11/518,292.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167.

Nina Dethlefs, Helen Hastie, Heriberto Cuayhuitl, and Oliver Lemon. 2013. Conditional random fields for responsive surface realisation using global features. In *In Proceedings of ACL*.

Milica Gašić, Dongho Kim, Pirros Tsiakoulis, Catherine Breslin, Matthew Henderson, Martin Szummer, Blaise Thomson, and Steve Young. 2014. Incremental on-line adaptation of pomdp-based dialogue managers to extended domains. In *In Proceedings on InterSpeech*.

Alex Graves, Marcus Liwicki, Horst Bunke, Jürgen Schmidhuber, and Santiago Fernández. 2008. Unconstrained on-line handwriting recognition with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 577–584.

Alex Graves, A-R Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Proceedings of IEEE Spoken Language Technology*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *CoRR*, abs/1404.2188.

Andrej Karpathy and Li Fei-Fei. 2014. Deep visual-semantic alignments for generating image descriptions. *CoRR*, abs/1412.2306.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical nlg framework for aggregated planning and realization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1406–1415, Sofia, Bulgaria, August. Association for Computational Linguistics.

Karen Kukich. 1987. Where do phrases come from: Some preliminary experiments in connectionist phrase generation. In *Natural Language Generation*, volume 135 of *NATO ASI Series*, pages 405–421. Springer Netherlands.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, ACL '98, pages 704–710.

Yann Lecun, León Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov.

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

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. 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.

Tomáš Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *In Proceedings on IEEE SLT workshop*.

Tomáš Mikolov, Martin Karafit, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *In Proceedings on InterSpeech*.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan H. Černocký, and Sanjeev Khudanpur. 2011a. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*.

Tomáš Mikolov, Stefan Kombrink, Anoop Deoras, Lukáš Burget, and Jan Černocký. 2011b. Rnnlm - recurrent neural network language modeling toolkit. In *In Proceedings on ASRU*.

Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Danilo Mirkovic and Lawrence Cavedon. 2011. Dialogue management using scripts, February 16. EP Patent 1,891,625.

Alice H. Oh and Alexander I. Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational Systems - Volume 3*, ANLP/NAACL-ConvSyst '00, pages 27–32.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, October.

Heather Pon-Barry, Karl Schultz, Elizabeth Owen Bratt, Brady Clark, and Stanley Peters. 2006. Responding to student uncertainty in spoken tutorial dialogue systems. *International Journal of Artificial Intelligence in Education*.

Adwait Ratnaparkhi. 2002. Trainable approaches to surface natural language generation and their application to conversational dialog systems. *Computer Speech and Language*. Spoken Language Generation.

Verena Rieser and Oliver Lemon. 2010. Natural language generation as planning under uncertainty for spoken dialogue systems. In *Empirical Methods in Natural Language Generation*, pages 105–120.

Tara N Sainath, A-r Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. 2013. Deep convolutional neural networks for lvcsr. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8614–8618. IEEE.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11):2673–2681.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentation in spoken dialog systems. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 79–86.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. 2014. Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14–25. Association for Computational Linguistics.

Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, New York, NY, USA. ACM.

Marilyn A Walker, Owen C Rambow, and Monica Rogati. 2002. Training a sentence planner for spoken dialogue using boosting. *Computer Speech and Language*, 16(3):409–433.

Wayne Ward and Sunil Issar. 1994. Recent improvements in the cmu spoken language understanding system. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 213–216. Association for Computational Linguistics.

Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, May.

Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680. Association for Computational Linguistics, October.