A Diversity-Promoting Objective Function for Neural Conversation Models

Jiwei Li[1*] **Michel Galley**[2] **Chris Brockett**[2] **Jianfeng Gao**[2] **Bill Dolan**[2]

[1]Stanford University, Stanford, CA, USA
jiweil@stanford.edu
[2]Microsoft Research, Redmond, WA, USA
{mgalley,chrisbkt,jfgao,billdol}@microsoft.com

Abstract

Sequence-to-sequence neural network models for generation of conversational responses tend to generate safe, commonplace responses (e.g., *I don't know*) regardless of the input. We suggest that the traditional objective function, i.e., the likelihood of output (responses) given input (messages) is unsuited to response generation tasks. Instead we propose using Maximum Mutual Information (MMI) as objective function in neural models. Experimental results demonstrate that the proposed objective function produces more diverse, interesting, and appropriate responses, yielding substantive gains in BLEU scores on two conversational datasets.

1 Introduction

Conversational agents are of growing importance in facilitating smooth interaction between humans and their electronic devices, yet traditional hand-crafted dialog (Levin et al., 2000; Young et al., 2010) poses major challenges for scalability and domain adaptation. Attention has thus turned to learning conversational patterns from data: researchers have begun to explore data-driven generation of conversational responses within the framework of statistical machine translation (SMT), either phrase-based (Ritter et al., 2010), or using neural networks to rerank, or directly in the form of sequence-to-sequence models (SEQ2SEQ) (Sordoni et al., 2015; Vinyals and Le, 2015; Shang et al., 2015; Serban et al., 2015; Wen et al., 2015). SEQ2SEQ models offer the promise of scalability and language-independence, together with the capacity to implicitly learn semantic and syntactic relations between pairs, and to capture contextual dependencies (Sordoni et al., 2015) in a way that was not possible with conventional SMT approaches (Ritter et al., 2011).

An engaging response generation system should be able to output grammatical, coherent responses that are diverse and interesting. In practice, however, neural conversation models exhibit a tendency to generate trivial or non-committal responses, often involving high-frequency phrases along the lines of *I don't know* or *I'm OK* (Sordoni et al., 2015; Serban et al., 2015; Vinyals and Le, 2015). Table 1 illustrates this phenomenon, showing top outputs from SEQ2SEQ models in terms of average log likelihood per token.[1] All the top-ranked responses are generic, along the lines of *I don't know* or *Oh my god!* Responses that seem more meaningful or specific can also be found in the N-best lists, but these rank much lower. In part at least, this behavior can be ascribed to the relative frequency of generic responses like *I don't know* in conversational datasets, in contrast with the relative sparsity of other, more contentful or specific alternative responses.[2] It appears that by optimizing for the likelihood of outputs/targets/responses given inputs/sources/messages, neural models assign high

* The entirety of this work was conducted at Microsoft.

[1]We trained a 4-layer SEQ2SEQ neural model with 1,000 dimensional units from 20 million conversation pairs in the OpenSubtitles database (OSDb).

[2]In our training dataset from the OpenSubtitles database (OSDb), 0.45% sentences contain the sequence *I don't know*, a high rate considering huge diversity of conversational content in the dataset.

Input (message)	Output (response)	$\log p$
What are you doing?	I don't know.	-0.862
	I don't know!	-1.035
	Nothing.	-1.062
	Get out of the way.	-1.088
	Get out of here.	-1.089
	I'm going home.	-1.092
	Oh my god!	-1.094
	I'm talking to you.	-1.100
What is your name?	I don't know.	-0.913
	I don't know!	-0.916
	I don't know, sir.	-0.920
	Oh, my god!	-0.974
	...	...
	My name is Robert.	-1.553
	My name is John.	-1.583
	My name's John.	-1.586
How old are you?	I don't know.	-0.794
	I'm fine.	-1.064
	I'm all right.	-1.167
	Don't know.	-1.169
	I'm not sure.	-1.170
	...	...
	Twenty-five.	-1.637
	Five.	-1.657
	Eight.	-1.713

Table 1: Responses generated by a standard SEQ2SEQ neural model trained on the OpenSubtitles dataset (60 million conversation pairs). Decoding is implemented with beam size set to 200. The top examples are the responses with the highest average probability log-likelihoods in the N-best list. Lower ranked, less generic responses have been manually chosen from the N-best list.

probability to "safe" responses. This objective function common in related tasks such as machine translation may thus be unsuited to generation tasks involving intrinsically diverse valid outputs.

The question is how to overcome the neural models' predilection for the commonplace. Intuitively, we want to capture not only the dependency of responses on messages, but also the inverse, the likelihood that a message will be provided to a given response. Whereas the sequence *I don't know* is of high probability in response to most question-related messages, the reverse will generally not be true.

We propose to capture this intuition by using Maximum Mutual Information (MMI), first introduced in speech recognition (Bahl et al., 1986; Brown, 1987), as an optimization objective that measures the mutual dependence between inputs and outputs. In this work, we present practical training and decoding strategies for neural generation models that use MMI as objective function. We demonstrate that using MMI results in a clear decrease in the proportion of generic response sequences produced by models trained and evaluated on two datasets–a large set of Twitter conversations and a subset of the OpenSubtitles movie database–generating correspondingly more varied and interesting outputs. We also find a significant performance boost from the proposed models as measured by BLEU (Papineni et al., 2002).

2 Related work

Earlier efforts to incorporate statistical methods into dialog systems typically relied on one of two approaches. The first is stochastic models built on top of hand-coded rules or templates (Levin et al., 2000; Young et al., 2010; Walker et al., 2003; Pieraccini et al., 2009; Wang et al., 2011). This approach is both expensive and difficult to extend to open-domain scenarios. The second approach also requires handcrafting, attempting to learn generation rules from a minimal set of authored rules or labels (Oh and Rudnicky, 2000; Ratnaparkhi, 2002; Banchs and Li, 2012; 0; Nio et al., 2014; Chen et al., 2013).

A third line of investigation–and the one that we adopt in this paper–was first introduced by Ritter et al. (2011), who framed the response generation task as a statistical machine translation (SMT) problem in which message-response phrase alignments are learned in an unsupervised manner from large volumes of conversational data. Inspired by recent progress in SMT stemming from the use of neural language models (Sutskever et al., 2014; Gao et al., 2014; Bahdanau et al., 2015; Luong et al., 2015), other research teams have subsequently attempted to extend these neural techniques to response generation. Sordoni et al. (2015) extend the system of Ritter et al. (2011) by re-ranking the N-best list of a phrasal SMT-based conversation system with a neural language model that incorporates the prior conversational context. There have also been a number of promising attempts to apply direct end-to-end neural encoding-decoding SEQ2SEQ models (Serban et al., 2015; Shang et al., 2015; Vinyals and Le, 2015; Wen et al., 2015). These SEQ2SEQ models are Long Short-Term Memory (LSTM) neural networks (Hochreiter and Schmidhuber, 1997) that are able to implicitly capture compositionality and long-span dependencies.

3 Sequence-to-Sequence Models

Given a sequence of inputs $X = \{x_1, x_2, ..., x_{n_X}\}$, an LSTM associates each time step with an input gate, a memory gate and an output gate, respectively denoted as i_t, f_t and o_t. We distinguish e and h where e_t denotes the vector for an individual text unit (for example, a word or sentence) at time step t while h_t denotes the vector computed by LSTM model at time t by combining e_t and h_{t-1}. c_t is the cell state vector at time t, and σ denotes the sigmoid function. Then, the vector representation h_t for each time step t is given by:

$$i_t = \sigma(W_i \cdot [h_{t-1}, e_t]) \tag{1}$$
$$f_t = \sigma(W_f \cdot [h_{t-1}, e_t]) \tag{2}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, e_t]) \tag{3}$$
$$l_t = \tanh(W_l \cdot [h_{t-1}, e_t]) \tag{4}$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \tag{5}$$
$$h_t^s = o_t \cdot \tanh(c_t) \tag{6}$$

where W_i, W_f, W_o, $W_l \in \mathbb{R}^{K \times 2K}$. In SEQ2SEQ generation tasks, each input X is paired with a sequence of outputs to predict: $Y = \{y_1, y_2, ..., y_{n_Y}\}$. The LSTM defines a distribution over outputs and sequentially predicts tokens using a softmax function:

$$p(Y|X) = \prod_{t=1}^{n_y} p(y_t|x_1, x_2, ..., x_t, y_1, y_2, ..., y_{t-1})$$
$$= \prod_{t=1}^{n_y} \frac{\exp(f(h_{t-1}, e_{y_t}))}{\sum_{y'} \exp(f(h_{t-1}, e_{y'}))}$$

where $f(h_{t-1}, e_{y_t})$ denotes the activation function between h_{t-1} and e_{y_t}, where h_{t-1} is the representation output from the LSTM at time $t-1$. Each sentence concludes with a special end-of-sentence symbol *EOS*. Commonly, the input and output each use different LSTMs with separate sets of compositional parameters to capture different compositional patterns.

During decoding, the algorithm terminates when an *EOS* token is predicted. At each time step, either a greedy approach or beam search can be adopted for word prediction. Greedy search selects the token with the largest conditional probability, the embedding of which is then combined with preceding output to predict the token at the next step.

4 MMI Models

4.1 Notation

In the response generation task, let S denote an input message sequence (source) $S = \{s_1, s_2, ..., s_{N_S}\}$ where N_S denotes the number of words in S. Let T (target) denote a sequence in response to source sequence S, where $T = \{t_1, t_2, ..., t_{N_T}, \textit{EOS}\}$, N_T is the length of the response (terminated by an *EOS* token) and w denotes a word token that is associated with a K dimensional distinct word embedding e_w. V denotes vocabulary size.

4.2 Objective Function

The standard objective function for sequence-to-sequence models is the log-likelihood of target T given source S, which at test time yields the optimization problem:

$$\hat{T} = \arg\max_T \left\{ \log p(T|S) \right\} \tag{7}$$

As discussed in the introduction, we surmise that this formulation leads to generic responses being generated, since it only selects for targets given sources, not the converse. To remedy this, we substitute Maximum Mutual Information (MMI) as objective function. In MMI, parameters are chosen to maximize (pairwise) mutual information between the source S and the target T:

$$\log \frac{p(S,T)}{p(S)p(T)} \tag{8}$$

It thus avoids favoring responses that unconditionally enjoy high probability, and instead biases towards those responses that are specific to the given input. The MMI objective can written as follows:[3]

$$\hat{T} = \arg\max_T \left\{ \log p(T|S) - \log p(T) \right\}$$

We use a generalization of the MMI objective which introduces a hyperparameter λ that controls how much to penalize generic responses:

$$\hat{T} = \arg\max_T \left\{ \log p(T|S) - \lambda \log p(T) \right\} \tag{9}$$

[3]Note: $\log \frac{p(S,T)}{p(S)p(T)} = \log \frac{p(T|S)}{p(T)} = \log p(T|S) - \log p(T)$

An alternate formulation of the MMI objective uses Bayes' theorem:

$$\log p(T) = \log p(T|S) + \log p(S) - \log p(S|T)$$

which lets us rewrite Equation 9 as follows:

$$\begin{aligned}\hat{T} &= \arg\max_{T} \big\{(1-\lambda)\log p(T|S) \\ &\qquad + \lambda \log p(S|T) - \lambda \log p(S)\big\} \\ &= \arg\max_{T} \big\{(1-\lambda)\log p(T|S) + \lambda \log p(S|T)\big\}\end{aligned} \tag{10}$$

This weighted MMI objective function can thus be viewed as representing a tradeoff between sources given targets (i.e., $p(S|T)$) and targets given sources (i.e., $p(T|S)$).

Although the MMI optimization criterion has been comprehensively studied for other tasks, such as acoustic modeling in speech recognition (Huang et al., 2001), adapting MMI to SEQ2SEQ training is empirically nontrivial. Moreover, we would like to be able to adjust the value λ in Equation 9 without repeatedly training neural network models from scratch, which would otherwise be extremely time-consuming. Accordingly, we did not train a joint model ($\log p(T|S) - \lambda \log p(T)$), but instead trained maximum likelihood models. The MMI criterion was used only during testing.

4.3 Practical Considerations

Responses can be generated either from Equation 9, i.e., $p(T|S) - \lambda p(T)$ or Equation 10 i.e., $(1-\lambda)p(T|S) + \lambda p(S|T)$. However, these strategies are difficult to apply directly to decoding since they can lead to ungrammatical responses (under Equation 9) or make decoding intractable (under Equation 10). In the rest of this section, we will discuss these issues and explain how we resolve them in practice.

4.3.1 $p(T|S) - \lambda p(T)$

The second term (i.e., $-\lambda p(T)$) functions as an anti-language model. It penalizes not only high-frequency, generic responses, but also fluent ones and thus can lead to ungrammatical outputs. In theory, this issue should not arise when λ is less than 1, since ungrammatical sentences should always be more severely penalized by the first term of the equation, i.e., $\log p(T|S)$. In practice, however, we found that the model tends to select ungrammatical outputs that escaped being penalized by $p(T|S)$.

Solution Let L_T be the length of target T. $p(T)$ in Equation 9 can be written as:

$$p(T) = \prod_{i=1}^{L_t} p(t_i|t_1, t_2, ..., t_{i-1}) \tag{11}$$

We replace the language model $p(T)$ with $U(T)$, which adapts the standard language model by multiplying by a weight $g(i)$ that is decremented monotonically as the index of the current token i increases:

$$U(T) = \prod_{i=1}^{L_t} p(t_i|t_1, t_2, ..., t_I) \cdot g(i) \tag{12}$$

The underlying intuition here is as follows: First, neural decoding combines the previously built representation with the word predicted at the current step. As decoding proceeds, the influence of the initial input on decoding (i.e., the source sentence representation) diminishes as additional previously-predicted words are encoded in the vector representations.[4] In other words, the first words to be predicted significantly determine the remainder of the sentence. Penalizing words predicted early on by the language model contributes more to the diversity of the sentence than it does to words predicted later. Second, as the influence of the input on decoding declines, the influence of the language model comes to dominate. We have observed that ungrammatical segments tend to appear in the latter part of the sentences, especially in long sentences.

We adopt the most straightforward form of $g(i)$ by by setting up a threshold (γ) by penalizing the first γ words where[5]

$$g(i) = \begin{cases} 1 & \text{if } i \leq \gamma \\ 0 & \text{if } i > \gamma \end{cases} \tag{13}$$

The objective Equation 9 can therefore be rewritten as:

$$\log p(T|S) - \lambda \log U(T) \tag{14}$$

where direct decoding is tractable.

[4] Attention models (Xu et al., 2015) may offer some promise of addressing this issue.

[5] We experimented with a smooth decay in $g(i)$ rather than a stepwise function, but this did not yield better performance.

4.3.2 $(1-\lambda)p(T|S) + \lambda p(S|T)$

Direct decoding from the above form is intractable, as the second part (i.e., $p(S|T)$) requires completion of target generation *before* $p(S|T)$ can be effectively computed. Due to the enormous search space for target T, exploring all possibilities is infeasible.

For practical reasons, then, we turn to an approximation approach that involves first generating N-best lists given the first part of objective function, i.e., standard SEQ2SEQ model $p(T|S)$. Then we rerank the N-best lists using the second term of the objective function. Since N-best lists produced by SEQ2SEQ models are generally grammatical, the final selected options are likely to be well-formed. Model reranking has obvious drawbacks. It results in non-globally-optimal solutions by first emphasizing standard SEQ2SEQ objectives. Moreover, it relies heavily on the system's success in generating a sufficiently diverse N-best set, requiring that a long list of N-best lists be generated for each message.

Despite these challenges, these two objectives work well in practice, significantly improving both the interestingness and the diversity of responses.

4.4 Training

Recent research has shown that deep LSTMs work better than single-layer LSTMs for SEQ2SEQ tasks (Vinyals et al., 2015; Sutskever et al., 2014; Li et al., 2015). We adopt a deep structure with four LSTM layers for encoding and four LSTM layers for decoding, each of which consists of a different set of parameters. Each LSTM layer consists of 1,000 hidden neurons, and the dimensionality of word embeddings is set to 1,000. Other training details are given below, broadly following Sutskever et al. (2014).

- LSTM parameters and word embeddings are initialized from a uniform distribution between [-0.08, 0.08].
- Stochastic gradient decent is implemented using a fixed learning rate of 0.1.
- Batch size is set to 256.
- Gradient clipping is adopted by scaling gradients when the norm exceeded a threshold of 1.

Our implementation on a single GPU processes at a speed of approximately 600-1200 tokens per second.[6]

[6]Tesla K40m, 1 Kepler GK110B, 2880 Cuda cores.

The $p(S|T)$ model described in Section 4.3.1 was trained using the same model as that of $p(T|S)$, with messages (S) and responses (T) interchanged.

4.5 Decoding

4.5.1 $p(T|S) - \lambda U(T)$

As described in Section 4.3.1, decoding using this model can be readily implemented by predicting tokens at each time-step. In addition, we found in our experiments that it is also important to take into account the length of responses in decoding. We thus linearly combine the loss function with length penalization, leading to an ultimate score for a given target T as follows:

$$Score(T) = p(T|S) - \lambda U(T) + \gamma L_T \tag{15}$$

where L_T denotes the length of the target and γ denotes associated weight. We optimize the length weight using grid search on N-best lists of response candidates. The N-best lists are generated using the decoder with beam size 200. We set a maximum length of 20 for generated candidates. N-best lists are then constructed so that sentences generated with an *EOS* token at each decoding time step are stored as decoding proceeds.

4.5.2 $(1-\lambda)p(T|S) + \lambda p(S|T)$

We generate N-best lists based on $P(T|S)$ and then rerank the list by linearly combining $p(T|S)$ with $\lambda p(S|T)$ and γL_T. We use grid search to tune the combination weights of λ and γ on the development set.

5 Experiments

5.1 Datasets

Twitter Conversation Triple Dataset We used an extension of the dataset described in Sordoni et al. (2015), which consists of 23 million conversational snippets randomly selected from a collection of 129M context-message-response triples extracted from the Twitter Firehose over the 3-month period from June through August 2012. For the purposes of our experiments, we limited context to the turn in the conversation immediately preceding the message. In our LSTM models, we used a simple input model in which contexts and messages are concatenated to form the source input.

For tuning and evaluation, we used the development dataset (2118 conversations) and the test dataset (2114 examples), augmented using IR to create a multi-reference set, as described by Sordoni et al. (2015). The selection criteria for these two datasets included a component of relevance/interestingness, with the result that dull responses will tend to be penalized in evaluation.

OpenSubtitles dataset In addition to unscripted Twitter conversations, we also used the OpenSubtitles dataset (Tiedemann, 2009), a large, noisy, open-domain dataset containing roughly 60M-70M scripted lines spoken by movie characters. This dataset does not specify which character speaks each subtitle line, which prevents us from inferring speaker turns. Following Vinyals et al. (2015), we make the simplifying assumption that each line of subtitle constitutes a full speaker turn. Our models are trained to predict the current turn given the preceding ones based on the assumption that consecutive turns belong to the same conversation. This introduces a degree of noise, since consecutive lines may not appear in the same conversation or scene, and may not even be spoken by the same character.

This limitation potentially renders the OSDb dataset unreliable for evaluation purposes. We therefore used data from the Internet Movie Script Database (IMSDB),[7] which explicitly identifies which character speaks each line of the script. This allowed us to identify consecutive message-response pairs spoken by different characters within the same scene. We randomly selected two subsets as development and test datasets, each containing 2,000 pairs, with source and target length restricted to the range of [6,18].

5.2 Evaluation

For parameter tuning and final evaluation, we used BLEU (Papineni et al., 2002), which was shown to correlate reasonably well with human judgment on the response generation task (Galley et al., 2015). In the case of the Twitter models, we used multi-reference BLEU. As the IMSDB data is too limited to support extraction of multiple references, only single reference BLEU was used in training and evaluating the OSDb models.

We did not follow Vinyals et al. (2015) in using perplexity as evaluation metric. Perplexity is unlikely to be a useful metric in our scenario, since our proposed model is designed to steer away from the standard SEQ2SEQ model in order diversify the outputs. Instead, we report degree of diversity by calculating the number of distinct unigrams and bigrams in generated responses. The value is scaled by total number of generated tokens to avoid favoring long sentences (shown as *distinct-1* and *distinct-2* in Tables 2 and 3).

5.3 Results

Twitter Dataset We first report performance on Twitter datasets in Table 2, along with results for different models (i.e., *Machine Translation* and *MT+neural reranking*) reprinted from Sordoni et al. (2015) on the same dataset.

Machine Translation is the phrase-based MT system described in (Ritter et al., 2011). MT features include commonly used ones in Moses (Koehn et al., 2007), e.g., forward and backward maximum likelihood translation probabilities, word and phrase penalties, linear distortion, etc. For more details, refer to Sordoni et al. (2015).

MT+neural reranking is the phrase-based MT system, reranked using neural models. N-best lists are first generated from the MT system. Recurrent neural models generate scores for N-best list candidates given the input messages. These generated scores are re-incorporated to rerank all the candidates. Additional features to score [1-4]-gram matches between context and response and between message and context (context and message match CMM features) are also employed, as in Sordoni et al. (2015).

MT+neural reranking achieves a BLEU score of 4.44, which corresponds, to the best of our knowledge, to the previous state-of-the-art performance on this Twitter dataset. Note that *Machine Translation* and *MT+neural reranking* are trained on a much larger dataset of roughly 50 million examples. A significant performance boost is observed from MMI over standard SEQ2SEQ, both in terms of BLEU score and diversity.

OpenSubtitles Dataset Our models achieve significantly lower BLEU scores on this dataset than

[7] IMSDB (`http://www.imsdb.com/`) is a relatively small database of around 0.4 million sentences and thus not suitable for open domain dialogue training.

Model	# of Training Data ($\approx$)	BLEU	*distinct-1*	*distinct-2*
SEQ2SEQ	23M	3.62	0.017	0.070
$p(T\|S) - \lambda U(T)$	23M	3.90 (+8.3%)	0.023 (+41.2%)	0.101 (+44.2%)
$(1-\lambda)p(T\|S) + \lambda p(S\|T)$	23M	**4.61** (+27.4%)	0.024 (+41.1%)	0.117 (+67.1%)
Machine Translation	50M	3.60	–	–
MT+neural reranking	50M	4.44	–	–

Table 2: Performance on the Twitter dataset of Standard SEQ2SEQ models and MMI models. *distinct-1* and *distinct-2* are respectively the number of distinct unigrams and bigrams divided by total number of generated words.

Model	BLEU	*distinct-1*	*distinct-2*
SEQ2SEQ	1.28	0.0056	0.0136
$p(T\|S)$ $-\lambda U(T)$	1.74 (+35.9%)	0.0184 (+228%)	0.066 (407%)
$(1-\lambda)p(T\|S)$ $+\lambda p(S\|T)$	1.44 (+28.2%)	0.0103 (+83.9%)	0.0303 (+122%)

Table 3: Performance on the OpenSubtitles dataset from Standard SEQ2SEQ models and MMI models.

those on Twitter dataset, primarily because the IMSDB data provides only single references for evaluation. We note, however, that standard SEQ2SEQ models yield lower levels of unigram diversity (*distinct-1*) on the OpenSubtitles dataset than on the Twitter data (0.0056 vs 0.017), which suggests that other factors may be in play. It is likely that movie dialogs may be much more concise and information-rich than typical conversations on Twitter, making it harder to match gold-standard responses and causing the learned models to strongly favor safe, conservative responses.

Table 3 shows that the $p(T|S) - \lambda U(T)$ model yields a significant performance boost, with a BLEU score increase of up to 36% and a more than 200% jump in unigram diversity. Our interpretation of this huge performance improvement is that the diversity and complexity of input messages lead standard SEQ2SEQ models to generate very conservative responses,[8] which fail to match more the interesting reference strings discussed typical of this dataset. This interpretation is also supported by the fact that the $(1-\lambda)p(T|S) + \lambda p(S|T)$ model does not produce as significant a performance boost as $p(T|S) - \lambda U(T)$. In the former, N-best lists generated using standard SEQ2SEQ models remain conservative and uninteresting, attenuating the impact of latter reranking. An important potential limitation of $(1-\lambda)p(T|S) + \lambda p(S|T)$ model is thus that its performance hinges on the initial generation of a highly diverse, informative N-best list.

[8]The strings *I don't know*, *I don't know what you are talking about*, *I don't think that is a good idea*, and *Oh my god* constitute 32% percent of all generated responses.

Input (message)	Output (response)
What are you doing?	I've been looking for you. I want to talk to you. Just making sure you're OK. I told you to shut up. Get out of here. I'm going home. I'm looking for a doctor.
What is your name?	John. Blue! Peter. Tyler. Daniel. My name is John. My name is Robert. My name's Daniel.
How old are you?	Twenty-five. Twenty-eight. Twenty-four. Long. Five. 15. Good. Eight.

Table 5: Examples generated by the $p(T|S) - \lambda U(T)$ model on the OpenSubtitles dataset.

Qualitative Evaluation Table 4 presents the top N-best candidates generated using the $p(T|S) - \lambda U(T)$ model for the messages described in Section 1. We see that MMI models generate significantly

message	SEQ2SEQ	MMI $P(T\|S) - \lambda P(T)$
I did not get the report from an mi-6 agent.	I don't think that's a good idea.	You did the right thing, did you?
You haven't been given an assignment in this case.	I don't know what you are talking about.	I've been looking all over for you.
I'm losing my grip.	I don't know what you are talking about.	I'm the only one in the world.
I am ready to help.	Come on, come on	I have something we need to talk about.
You programmed me to gather intelligence. That's all I've ever done.	You know that, don't you?	You do have fun, don't you?
I mean, we'd have to talk to him.	I mean, I don't know.	I mean, he's a good guy.
This has nothing to do with him.	You don't know anything about him, do you?	I know him as much.
How come you never say it?	I don't know	Because I don't want to hurt you
He is the one behind this scheme to crash x-net.	I don't know what you are talking about	Yes, he is, isn't he?
How much time do you have here?	I don't know.	Not long enough. Sorry, sir.

Table 4: Responses from the standard SEQ2SEQ model and $p(T|S) - \lambda U(T)$ models on the OpenSubtitles dataset.

message	SEQ2SEQ	MMI $(1-\lambda)p(T\|S) + \lambda p(S\|T)$
Ohh I've never seen that! How long does it take you guys to learn the drill?	I don't even know.	Like 2 weeks ago!!
You'll know when we sign Heskey in 2 weeks	I'll see what I can do	I sure hope so..
I'm waiting!! I have even got biscuits waiting for you??	I have no idea what you're talking about	I'll send you a pic when I get home xx
My parents actually had touch too much played at their wedding.	I have no idea what you're talking about	How old are they??
Say anything you want... I know you love me and I love you.	I love you so much.	I love you too, baby.
I am off all week next week	What time you going?	What time you going?
How are you doing?	I'm good, thanks.	I'm good, you?

Table 6: Responses from the standard SEQ2SEQ model and $(1-\lambda)p(T|S) + \lambda p(S|T)$ models from the Twitter dataset.

more interesting outputs than standard SEQ2SEQ models.

In Table 5, we present responses generated by different models. All examples were randomly sampled (without cherry picking). We see that standard SEQ2SEQ models tend to generate reasonable responses to simple messages such as *How are you doing?* or *I love you*. As the complexity of the message increases, however, the outputs switch to more conservative and duller forms, such as *I don't know* or *I don't know what you are talking about*. An occasional answer of this kind might go unnoticed in a natural conversation, but a dialog agent that *always* produces such responses risks being perceived as uncooperative or even rude. MMI models produce far more diverse and interesting responses, though they can also occasionally generate completely irrelevant outputs, e.g., **message**: *I am losing my grip*; **response**: *I'm the only one in the world*. To mitigate this problem, we will need to seek a better trade-off that will permit generation of responses that are both interesting and relevant.

Phrase-based MT versus Neural Generation Neural models using MMI as objective function outperform MT in BLEU, establishing a new state-of-the-art result on the Twitter conversational dataset. More than that, they address several limitations inherent in the MT framework. First, neural models are more flexible in leveraging contextual information such as speaker characteristics, specific topics, domain information, and scenarios that are related to the dialogue. Second, these models are more scalable. Instead of relying on a big phrase translation table to memorize individual response pairs, they encode large amounts of contextual information using a low-dimensionality vector so that semantically similar messages lead to similar responses. Finally, neural models allow end-to-end optimization of model parameters, yielding significant performance gains over earlier methods.

6 Conclusions

We investigated an issue encountered when applying SEQ2SEQ models to conversational response generation: These models tend to generate safe, commonplace responses (e.g., *I don't know*) regardless of the input. Our analysis suggests that the issue is at least in part attributable to the use of the traditional objective function, namely the unidirectional likelihood of output (responses) given input (messages), widely used in Statistical Machine Translation and other machine learning models. To remedy this problem, we have proposed using Maximum Mutual Information (MMI) as the objective function in neural models, in order to capture not only the dependency of responses on messages but also the inverse. Experimental results demonstrate that the proposed MMI models produce more diverse, interesting, and appropriate responses, yielding substantive gains in BLEU scores on two conversational datasets.

To the best of our knowledge, this paper represents the first work to address the issue of output diversity in the neural generation framework. We have focused on the algorithmic dimensions of the problem. Unquestionably numerous other factors such as grounding, persona (of both user and agent), and intent also play a significant role in generating diverse, conversationally interesting outputs, but those must be left for future investigation. The implications of this work extend beyond conversational response generation, since the challenge of producing interesting outputs also arises in other neural generation tasks, including image-description generation and question answering, and potentially any task where mutual correspondences must be modeled.

References

David Ameixa, Luisa Coheur, Pedro Fialho, and Paulo Quaresma. 2014. Luke, I am your father: dealing with out-of-domain requests by using movies subtitles. In *Intelligent Virtual Agents*, pages 13–21. Springer.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of the International Conference on Learning Representations (ICLR)*.

L. Bahl, P. Brown, P. de Souza, and R. Mercer. 1986. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. pages 49–52.

Rafael E Banchs and Haizhou Li. 2012. IRIS: a chat-oriented dialogue system based on the vector space model. In *Proc. of the ACL 2012 System Demonstrations*, pages 37–42. Association for Computational Linguistics.

Peter F. Brown. 1987. *The Acoustic-modeling Problem in Automatic Speech Recognition*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

Yun-Nung Chen, Wei Yu Wang, and Alexander Rudnicky. 2013. An empirical investigation of sparse log-linear models for improved dialogue act classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8317–8321. IEEE.

Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. 2015. ΔBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. In *Proc. of ACL-IJCNLP*, pages 445–450, Beijing, China, July.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2014. Learning continuous phrase representations for translation modeling. In *Proc. of ACL*, pages 699–709, Baltimore, Maryland. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Foreword By-Reddy. 2001. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice Hall.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *Speech and Audio Processing, IEEE Transactions on*, 8(1):11–23.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proc. of ACL-IJCNLP*, pages 11–19, Beijing, China.

Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, Mirna Adriani, and Satoshi Nakamura. 2014. Developing non-goal dialog system based on examples of drama television. In *Natural Interaction with Robots, Knowbots and Smartphones*, pages 355–361. Springer.

Alice H Oh and Alexander I Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *Proc. of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*, pages 27–32. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

Roberto Pieraccini, David Suendermann, Krishna Dayanidhi, and Jackson Liscombe. 2009. Are we there yet? research in commercial spoken dialog systems. In *Text, Speech and Dialogue*, pages 3–13. Springer.

Adwait Ratnaparkhi. 2002. Trainable approaches to surface natural language generation and their application to conversational dialog systems. *Computer Speech & Language*, 16(3):435–455.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations.

Alan Ritter, Colin Cherry, and William Dolan. 2011. Data-driven response generation in social media. In *Proc. of EMNLP*, pages 583–593. Association for Computational Linguistics.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *ACL-IJCNLP*, pages 1577–1586.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Meg Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proc. of NAACL-HLT*.

Ilya Sutskever, Oriol Vinyals, and Quoc Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jörg Tiedemann. 2009. News from OPUS – a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proc. of ICML Deep Learning Workshop*.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Proc. of NIPS*.

Marilyn A Walker, Rashmi Prasad, and Amanda Stent. 2003. A trainable generator for recommendations in multimodal dialog. In *INTERSPEECH*.

William Yang Wang, Ron Artstein, Anton Leuski, and David Traum. 2011. Improving spoken dialogue understanding using phonetic mixture models. In *FLAIRS Conference*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proc. of EMNLP*, pages 1711–1721, Lisbon, Portugal, September. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In David Blei and Francis Bach, editors, *Proc. of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2048–2057. JMLR Workshop and Conference Proceedings.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.