A Diversity-Promoting Objective Function for Neural Conversation Models

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

1Stanford University, Stanford, CA, USA
jiweil@stanford.edu
2Microsoft 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 (response) given input (message) is unsuited to response generation tasks. Instead we propose using Maximum Mutual Information (MMI) as the objective function in neural models. 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 and in human evaluations.

1 Introduction

Conversational agents are of growing importance in facilitating smooth interaction between humans and their electronic devices, yet conventional dialog systems continue to face major challenges in the form of robustness, 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., 2011), or using neural networks to rerank, or directly in the form of sequence-to-sequence (SEQ2SEQ) models (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 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 tend 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. All the top-ranked responses are generic. Responses that seem more meaningful or specific can also be found in the N-best lists, but 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 more contentful alternative responses.1 It appears that by optimizing for the likelihood of outputs given inputs, neural models assign high probability to “safe” responses. This objective function, common in related tasks such as machine translation, may be unsuited to generation tasks involving intrinsically diverse outputs. Intuitively, it seems desirable to take into account not only the dependency of responses on messages, but also the inverse, the likelihood that a message will be provided to a given response.

We propose to capture this intuition by using Maximum Mutual Information (MMI), first introduced in

---

1In 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 this dataset.
speech recognition (Bahl et al., 1986; Brown, 1987), as an optimization objective that measures the mutual dependence between inputs and outputs. Below, we present practical strategies for neural generation models that use MMI as an objective function. We show that use of MMI results in a clear decrease in the proportion of generic response sequences, generating correspondingly more varied and interesting outputs.

2 Related work
Earlier efforts to build statistical methods into dialog systems have 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 attempts 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), which also implies handcrafting.

A newer line of investigation, introduced by Ritter et al. (2011), frames response generation as a statistical machine translation (SMT) problem. 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) has inspired attempts to extend these neural techniques to response generation. Sordoni et al. (2015) improved upon Ritter et al. (2011) by rescoring the output of a phrasal SMT-based conversation system with a SEQ2SEQ model that incorporates prior context. There have other attempts to apply SEQ2SEQ models (Serban et al., 2015; Shang et al., 2015; Vinyals and Le, 2015; Wen et al., 2015) using Long Short-Term Memory (LSTM) neural networks (Hochreiter and Schmidhuber, 1997), which can implicitly capture compositionality and long-span dependencies.

Prior work in generation has sought to increase diversity, but with different goals and techniques. Carbonell and Goldstein (1998) and Gimpel (2013) produce multiple outputs that are mutually diverse, either non-redundant summary sentences or N-best lists. Our goal, however, is to produce a single non-trivial output, and our method does not require identifying lexical overlap to foster diversity.2

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 \( \sigma \) denotes the sigmoid function. Then, the vector representation \( h_t \) for each time step \( t \) is given by:

\[
\begin{align*}
i_t &= \sigma(W_i \cdot [h_{t-1}, e_t]) \\
f_t &= \sigma(W_f \cdot [h_{t-1}, e_t]) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, e_t]) \\
l_t &= \tanh(W_l \cdot [h_{t-1}, e_t]) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot l_t \\
h_t &= o_t \cdot \tanh(\sigma_t)
\end{align*}
\]

2Augmenting our technique with MMR-based (Carbonell and Goldstein, 1998) diversity helped increase lexical but not semantic diversity (e.g., I don’t know vs. I haven’t a clue), and with no gain in performance.
where \(W_t, W_f, W_o, W_l \in \mathbb{R}^{K \times 2K}\). 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, input and output use different LSTMs with separate 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 \(T = \{t_1, t_2, ..., t_{N_T}, EOS\}\), \(N_T\) is the length of the response (terminated by an \(EOS\) token) and \(t\) denotes a word token that is associated with a \(K\) dimensional distinct word embedding \(e_t\). \(V\) denotes vocabulary size.

4.2 MMI Criterion

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 statistical decision problem:

\[
\hat{T} = \arg \max_T \{ \log p(T|S) \}
\]  

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 replace it with Maximum Mutual Information (MMI) as the objective function. In MMI, parameters are chosen to maximize (pairwise) mutual information between the source \(S\) and the target \(T\):

\[
\log \frac{p(S, T)}{p(S)p(T)}
\]

This 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:

\[
\hat{T} = \arg \max_T \{ \log p(T|S) - \log p(T) \}
\]

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

\[
\hat{T} = \arg \max_T \{ (1 - \lambda) \log p(T|S) - \lambda \log p(T) \}
\]

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:

\[
\hat{T} = \arg \max_T \{ (1 - \lambda) \log p(T|S) + \lambda \log p(S|T) \}
\]

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 \(\lambda\) 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, and used the MMI criterion only during testing.

\[\text{Note: } \log \frac{p(S, T)}{p(S)p(T)} = \log \frac{p(T|S)}{p(T)} = \log p(T|S) - \log p(T)\]
4.3 Practical Considerations

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

4.3.1 MMI-antiLM

The second term of \( \log p(T|S) - \lambda \log p(T) \) 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 \( \lambda \) 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:

\[
p(T) = \prod_{i=1}^{L_T} p(t_i|t_1, t_2, ..., t_{i-1})
\] (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:

\[
U(T) = \prod_{i=1}^{L_T} p(t_i|t_1, t_2, ..., t_{i-1}) \cdot g(i)
\] (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 (\( \gamma \)) by penalizing the first \( \gamma \) words where\(^5\)

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

The objective Equation 9 can thus be rewritten as:

\[
\log p(T|S) - \lambda \log U(T)
\] (14)

where direct decoding is tractable.

4.3.2 MMI-bidi

Direct decoding from \( (1 - \lambda) \log p(T|S) + \lambda \log p(S|T) \) 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.

Nevertheless, these two variants of the MMI criterion work well in practice, significantly improving both the interestingness and the diversity of responses.

\(^4\)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.
4.4 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). 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 aligned with Sutskever et al. (2014).

- LSTM parameters and embeddings are initialized from a uniform distribution in [-0.08, 0.08].
- Stochastic gradient descent 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.\(^6\)

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 MMI-antiLM

As described in Section 4.3.1, decoding using \(\log p(T|S) - \lambda U(T)\) 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 an ultimate score for a given target \(T\) as follows:

\[
Score(T) = p(T|S) - \lambda U(T) + \gamma L_T
\]

where \(L_T\) denotes the length of the target and \(\gamma\) denotes associated weight. We optimize \(\gamma\) and \(\lambda\) using MERT (Och, 2003) 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 MMI-bidi

We generate N-best lists based on \(P(T|S)\) and then rerank the list by linearly combining \(P(T|S)\), \(\lambda p(S|T)\), and \(\gamma L_T\). We use MERT to tune the weights \(\lambda\) and \(\gamma\) 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 information retrieval methods 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 (OSDb) 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. For evaluation purposes, we therefore used data from the In-
Table 2: Performance on the Twitter dataset of 4-layer 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.

Table 3: Performance on the OpenSubtitles dataset for the SEQ2SEQ baseline and two MMI models.

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. The baseline is the SEQ2SEQ model with its standard likelihood objective and a beam size of 200. We compare this baseline against greedy-search SEQ2SEQ (Vinyals and Le, 2015), which achieves higher diversity by increasing search errors.

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 to the best of our knowledge represents the previous state-of-the-art performance on this Twitter dataset. Note that Machine Translation and MT+neural reranking are trained on a much

---

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.
larger dataset of roughly 50 million examples. A significant performance boost is observed from MMI-bidi over baseline SEQ2SEQ, both in terms of BLEU score and diversity.

OpenSubtitles Dataset All models achieve significantly lower BLEU scores on this dataset than on the Twitter dataset, primarily because the IMSDB data provides only single references for evaluation. We note, however, that baseline 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 are 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 MMI-antiLM 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 typical of this dataset. This interpretation is also supported by the fact that the MMI-bidi model does not produce as significant a performance boost as MMI-antiLM. In the case of MMI-bidi, N-best lists generated using standard SEQ2SEQ models remain conservative and uninteresting, attenuating the impact of later reranking. An important potential limitation of MMI-bidi model is thus that its performance hinges on the initial generation of a highly diverse, informative N-best list.

Qualitative Evaluation We employed crowdsourced judges to provide evaluations for a random sample of 1000 items in the Twitter test dataset. Table 6 shows the results of human evaluations between paired systems. Each output pair was ranked by 5 judges, who were asked to decide which of the two outputs was better. They were instructed to prefer outputs that were more specific (relevant) to the message and preceding context, as opposed to those that were more generic. Ties were permitted. Identical strings were algorithmically assigned the same score. The mean of differences between outputs is shown as the gain for MMI-bidi over the competing system. At a significance level of \(\alpha = 0.05\), we find that MMI-bidi outperforms both baseline and greedy SEQ2SEQ systems, as well as the weaker SMT and SMT+RNN baselines. MMI-bidi outperforms SMT in human evaluations despite the greater lexical diversity of MT output.

Separately, judges were also asked to rate overall quality of MMI-bidi output over the same 1000-item sample in isolation, each output being evaluated by 7 judges in context using a 5-point scale. The mean rating was 3.84 (median: 3.85, 1st Qu: 3.57, 3rd Qu: 4.14), suggesting that overall MMI-bidi output does appear reasonably acceptable to human judges.

Table 7 presents the N-best candidates generated using the MMI-bidi model for the inputs of Table 1. We see that MMI generates significantly more interesting outputs than SEQ2SEQ.

In Tables 4 and 5, we present responses generated by different models. All examples were randomly sampled (without cherry picking). We see that the

---

\(^8\)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.

---

<table>
<thead>
<tr>
<th>Comparator</th>
<th>Gain</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMT (Ritter et al., 2011)</td>
<td>0.29</td>
<td>[0.25, 0.32]</td>
</tr>
<tr>
<td>SMT+neural reranking</td>
<td>0.28</td>
<td>[0.25, 0.32]</td>
</tr>
<tr>
<td>SEQ2SEQ (baseline)</td>
<td>0.11</td>
<td>[0.07, 0.14]</td>
</tr>
<tr>
<td>SEQ2SEQ (greedy)</td>
<td>0.08</td>
<td>[0.04, 0.11]</td>
</tr>
</tbody>
</table>

Table 6: MMI-bidi gains over comparator systems, based on pairwise human judgments.

<table>
<thead>
<tr>
<th>Input: What are you doing?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I’ve been looking for you.</td>
</tr>
<tr>
<td>2. I want to talk to you.</td>
</tr>
</tbody>
</table>

Input: What is your name?
2. Peter. | 5. My name is John. |
3. Tyler. | 6. My name is Robert. |

Input: How old are you?
1. Twenty-eight. | 4. Five. |
2. Twenty-four. | 5. 15. |

Table 7: Examples generated by the MMI-antiLM model on the OpenSubtitles dataset.
baseline SEQ2SEQ model tends 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, 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. MMI-bidi models, on the other hand, produce far more diverse and interesting responses.

### 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 unidirectional likelihood of output (responses) given input (messages). To remedy this, we have proposed using Maximum Mutual Information (MMI) as the objective function. Our results demonstrate that the proposed MMI models produce more diverse and interesting responses, while improving quality as measured by BLEU and human evaluation.

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 role in generating diverse, conversationally interesting outputs. These must be left for future investigation. Since the challenge of pro-

| 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 SEQ2SEQ baseline and MMI-antiLM models on the OpenSubtitles dataset.

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

Table 5: Responses from the SEQ2SEQ baseline and MMI-bidi models on the Twitter dataset.
producing interesting outputs also arises in other neural generation tasks, including image-description generation, question answering, and potentially any task where mutual correspondences must be modeled, the implications of this work extend well beyond conversational response generation.

References


Roberto Pieraccini, David Suendermann, Krishna Dayanidhi, and Jackson Liscombe. 2009. Are we there yet? research in commercial spoken dialog


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.