Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models

Louis Shao[1]*, Stephan Gouws[3]*, Denny Britz[2]†, Anna Goldie[2], Brian Strope[1], Ray Kurzweil[1]
{overmind,sgouws,dennybritz,agoldie,bps,raykurzweil}@google.com
[1]Google Research and [2]Google Brain
Mountain View, CA, USA
and
[3]Google Brain
London, UK

Abstract

Sequence-to-sequence models have been applied to the conversation response generation problem where the source sequence is the conversation history and the target sequence is the response. Unlike translation, conversation responding is inherently creative. The generation of long, informative, coherent, and diverse responses remains a hard task. In this work, we focus on the single turn setting. We add self-attention to the decoder to maintain coherence in longer responses, and we propose a practical approach, called the glimpse-model, for scaling to large datasets. We introduce a stochastic beam-search algorithm with segment-by-segment reranking which lets us inject diversity earlier in the generation process. We trained on a combined data set of over 2.3B conversation messages mined from the web. In human evaluation studies, our method produces longer responses overall, with a higher proportion rated as acceptable and excellent as length increases, compared to baseline sequence-to-sequence models with explicit length-promotion. A back-off strategy produces better responses overall, in the full spectrum of lengths.

1 Introduction

Building computer systems capable of general-purpose conversation is a challenging problem. However, it is a necessary step toward building intelligent agents that can interact with humans via natural language, and for eventually passing the Turing test. The sequence-to-sequence (*seq2seq*) model has proven very popular as a purely data-driven approach in domains that can be cast as learning to map to and from variable-length sequences, with state-of-the art results in many domains, including machine translation (Cho et al., 2014; Sutskever et al., 2014; Wu et al., 2016). Neural conversation models are the latest development in the domain of conversation modeling, with the promise of training computers to converse in an end-to-end fashion (Vinyals and Le, 2015; Shang et al., 2015; Sordoni et al., 2015; Wen et al., 2016). Despite promising results, there are still many challenges with this approach. In particular, these models produce short, generic responses that lack diversity (Sordoni et al., 2015; Li et al., 2015). Even when longer responses are explicitly encouraged (e.g. via length normalization), they tend to be incoherent (*"The sun is in the center of the sun."*), redundant (*"i like cake and cake"*), or contradictory (*"I don't own a gun, but I do own a gun."*).

In this paper, we provide two methods to address these issues with minimal modifications to the standard seq2seq model. First, we present a *glimpse model* that only trains on fixed-length segments of the target-side at a time, allowing us to scale up training to larger data sets. Second, we introduce a *segment-based stochastic decoding technique* which injects diversity earlier in the generated responses. Together, we find that these two methods lead to both longer responses and higher ratings, compared to a baseline seq2seq model with explicit length and diversity-promoting heuristics integrated into the generation procedure (see Table 1 for examples generated using our model).

In Section 2, we present a high-level overview of these two techniques. We then discuss each

[*] Both authors contributed equally to this work.

[†] Work done as a member of the Google Brain Residency program (g.co/brainresidency).

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2193–2202
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

technique in more detail in Sections 3 and 4. Finally, we report small and large-scale experimental evaluations of the proposed techniques in Section 5.

2 Overview and Motivation

A major difference between translation and responding to conversations is that, in the former, the high-level semantic content to generate in the target sequence $\mathbf{y}$ is completely given by the source sequence, *i.e.*, given the source $\mathbf{x}$, there is low conditional entropy in the target distribution $P(\mathbf{y}|\mathbf{x})$. In the seq2seq approach, the decoder network therefore only has to keep track of where it is in the output, and the content to generate can be transformed from the relevant parts in the source via the attention mechanism (Bahdanau et al., 2014). In contrast, in conversation response generation, the prompt turn may be short and general (e.g., *"what do you have planned tonight"*), while an appropriate response may be long and informative.

The standard seq2seq model struggles with generating long responses, since the decoder has to keep track of everything output so far in its fixed-length hidden state vector, which leads to incoherent or even contradictory outputs. To combat this, we propose to integrate *target-side attention* into the decoder network, so it can keep track of what has been output so far. This frees up capacity in the hidden state for modeling the higher-level semantics required during the generation of coherent longer responses. We were able to achieve small perplexity gains using this idea on the small OpenSubtitles 2009 data set (Tiedemann, 2009). However, we found it to be too memory-intensive when scaling up to larger data sets.

As a trade-off, we propose a technique (called the 'glimpse model') which interpolates between source-side-only attention on the encoder, and source and target-side attention on the encoder and decoder, respectively. Our solution simply trains the decoder on fixed-length *glimpses* from the target side, while having both the source sequence and the part of the target sequence before the glimpse on the encoder, thereby sharing the attention mechanism on the encoder. This can be implemented as a simple data-preprocessing technique with an unmodified standard seq2seq implementation, and allows us to scale training to very large data sets without running into any memory issues. See Figure 1 for a graphical overview, where we illustrate this idea with a glimpse-model of length 3.

Given such a trained model, the next challenge is how to generate long, coherent, and diverse responses with the model. As observed in the previous section and in other work, standard maximum a posteriori (MAP) decoding using beam search often yields short, uninformative, and high-frequency responses. One approach to produce longer outputs is to employ length-promoting heuristics (such as length-normalization (Wu et al., 2016)) during decoding. We find this increases the length of the outputs, however often at the expense of coherence. Another approach to explicitly create variation in the generated responses is to rerank the N-best MAP-decoded list of responses from the model using diversity-promoting heuristics (Li et al., 2015) or a backward RNN (Wen et al., 2015). We find this works for shorter responses, but not for long responses, primarily for two reasons: First, the method relies on the MAP-decoding to produce the N-best list, and as mentioned above, MAP-decoding prefers short, generic responses. Second, it is too late to delay reranking in the beam search until the whole sequence has been generated, since beam-search decoding tends to yield beams with low diversity per given prompt, even when the number of beams is high. Instead, our solution is to break up the reranking over shorter segments, and to rerank segment-by-segment, thereby injecting diversity earlier during the decoding process, where it has the most impact on the resulting diversity of the generated beams.

To further improve variation in the generated responses, we replace the deterministic MAP-decoding of the beam search procedure with sampling. If a model successfully captures the distribution of responses given targets, one can expect simple greedy sampling to produce reasonable responses. However, due to model underfitting, the learned distributions are often not sharp enough, causing step-by-step sampling to accumulate errors along the way, manifesting as incoherent outputs. We find that integrating sampling into the beam-search procedure yields responses that are more coherent and with more variation overall.

In summary, the contributions of this work are the following:

1. We propose to integrate target-side attention in neural conversation models, and provide a

PROMPT:	**RESPONSE FROM BASELINE (A) vs OUR METHOD (B)**
what do you have planned tonight	**A:** I have no idea what I want to do with my life. I have no idea what I want to (Bad, Mediocre, Bad) **B:** I was planning on doing a few of these, but I'm not sure what to choose. I need to find a place to camp, and I don't have a lot of time to make a plan. (Excellent, Good, Mediocre)
I'd like to read a good science fiction novel. Haven't read scifi in a while.	**A:** If you like sci fi you should check out The Name of the Wind by Patrick Rothfuss (Excellent, Excellent, Excellent) **B:** If you want to know what sci-fi is, read the first book by Neil Gaiman. It's a sci-fi sci-fi sci-fi novel. It is really good and I recommend it to all of you. It has a great premise. (Acceptable, Excellent, Acceptable)
Got any big weekend plans?	**A:** I have plans for the weekend but I don't know if I will be able to make it to the (Bad, Mediocre, Bad) **B:** I don't really have much to do on Saturday. But I plan on going to the movies at the mall and going to the show. (Good, Good, Bad)
Sometimes flying messes my ears, because of the air pressure. Especially if I have a cold or something, flying gets me messed up.	**A:** I don't know why, but when I was a kid I used to do this all the time when I was younger. (Good, Bad, Bad) **B:** I have no idea why, but I feel exactly the same way. I don't know why, but my brain is so sensitive to air that it makes my nose twitch. (Excellent, Mediocre, Good)
Yeah. I have this friend, and we get bored just hanging out here sometimes, so we take little trips to places. We went to Seattle a couple of months ago.	**A:** What kind of things do you like to do? (Excellent, Excellent, Excellent) **B:** That sounds like a great time. I have a lot of friends that go to Seattle, so I was wondering if you could share any of the locations? (Excellent, Excellent, Excellent)
have you seen Suicide Squad?	**A:** I have not, but I will check it out. (Good, Good, Good) **B:** I saw it in a theater once and it was great. It was very... I don't know, a little dark. I like the psychological effects and the way it portrays the characters in the movie. (Excellent, Excellent, Excellent)
Actually I like the snow too, except for today. It's all slushy and it's all over the sidewalks and it's gross.	**A:** I hate the snow. I hate the snow. I hate the snow. I hate the snow. I hate the snow. I hate the snow. I hate the snow. I hate the snow. (Acceptable, Good, Good) **B:** I'm in the same boat as you. I hate it when it's snowy and I don't like to shovel it all day. :(So this is a perfect excuse to not get out of bed and go somewhere else. (Excellent, Good, Acceptable)
What do you hope to see in the future?	**A:** I want to be able to go back in time to when I was born. I want to be (Mediocre, Bad, Mediocre) **B:** I want to see an actual movie where the characters have the ability to hide the fact that they are actually in the movie. (Good, Acceptable, Acceptable)
what dangers could happen with the singularity?	**A:** I don't think it would be that big of a deal. (Acceptable, Mediocre, Acceptable) **B:** If you look at it from the perspective of an alien, you can't say that there aren't any alternative ways to deal with it. (Acceptable, Mediocre, Acceptable)
I think you're a little crazy.	**A:** I don't think you know what that word means. (Acceptable, Good, Good) **B:** I am, actually. I don't think I'm crazy though. I certainly am. (Acceptable, Mediocre, Acceptable)

Table 1: Some example responses generated from the baseline (**A**) compared to our model (**B**), with human ratings in parentheses. These were selected for quality. See the appendix for an unbiased sample.

practical approach, referred to as the **glimpse model**, which scales well and is easy to implement on top of the standard sequence-to-sequence model.

2. We introduce a **stochastic beam-search procedure** with segment-by-segment reranking which improves the diversity of the generated responses.

3. We present large-scale experiments with human evaluations showing the proposed techniques improve over strong baselines.

4. We release our collection of context-free conversation prompts used in our evaluations as a benchmark for future open-domain conversation response research.

3 Seq2Seq Model with Attention on Target

We discuss conversation response generation in the sequence-to-sequence problem setting. In this setting, there is a source sequence $\mathbf{x} = (x_1, x_2, ..., x_M)$, and a target sequence $\mathbf{y} = (y_0, y_1, y_2, ..., y_N)$. We assume y_0 is always the start-of-sequence token and y_N is the end-of-sequence token. In a typical sequence-to-sequence model, the encoder gets its input from the source sequence $\mathbf{x}$ and the decoder models the conditional language model $P(\mathbf{y}|\mathbf{x})$ of the target sequence $\mathbf{y}$, given $\mathbf{x}$.

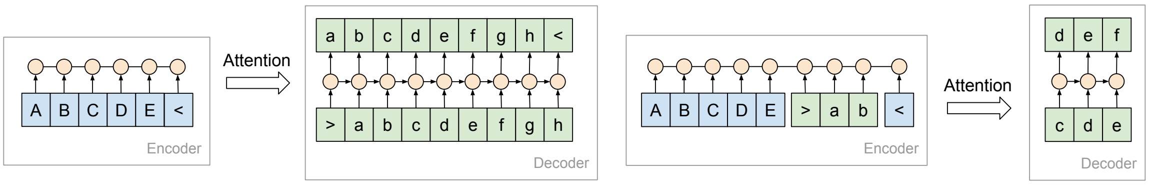
Seq2seq models with attention (Bahdanau et al., 2014) parameterize the per-symbol conditional probability as:

$$P\left(y_i|\mathbf{y}_{[0:i-1]};\mathbf{x}\right) = \mathrm{DecoderRNN}\,(\; y_{i-1}, h_{i-1}, \mathrm{Attention}\left(h_{i-1}, \mathbf{x}\right)) \tag{1}$$

for $1 \leq i \leq N$, where DecoderRNN() is a recurrent neural network that map the sequence of decoder symbols into fixed-length vectors, and Attention() is a function that yields a fixed-size vector summary of the encoder symbols $\mathbf{x}$ (the 'focus') most relevant to predicting y_i, given the previous recurrent state of the network h_{i-1} (the 'context'). The full conditional probability follows from the product rule, as:

$$P\left(\mathbf{y}|\mathbf{x}\right) = \prod_{i=1}^{N} P\left(y_i|\mathbf{y}_{[0:i-1]};\mathbf{x}\right) \tag{2}$$

We propose to implement **target-side attention** by augmenting the attention mechanism to include the part of the target sequence already generated, *i.e.*, we include $\mathbf{y}_{[0:i-2]}$ in the arguments to the attention function: Attention$(h_{i-1}, \mathbf{y}_{[0:i-2]}, \mathbf{x})$. We implemented this in TensorFlow (Abadi et al., 2015) using 3 LSTM layers on both the encoder and the decoder, with 1024 units per layer. We experimented on the OpenSubtitles 2009 data set, and obtained a small perplexity gain from the target-side attention: 24.6 without versus 24.2 with. However, OpenSubtitles is a small data set,



(a) The vanilla sequence-to-sequence model.

(b) Length-3 Target-glimpse Model

Figure 1: The vanilla seq2seq with attention on the left, and our proposed target-glimpse model on the right. The symbol “>” and “<” are start-of-sequence and end-of-sequence, respectively.

and the majority of its response sequences are shorter than 10 tokens. This may prevent us from seeing bigger gains, since our method is designed to help with longer outputs. In order to train on the much larger Reddit data set, we implemented this method on top of the GNMT model (Wu et al., 2016). Unfortunately, we met with frequent out-of-memory issues, as the 8-layer GNMT model is already very memory-intensive, and adding target-side attention made it even more so. Ideally, we would like to retain the model's capacity in order to train a rich response model, and therefore a more efficient approach is necessary.

To this end, we propose the **target-glimpse model** which has a fixed-length decoder. The target-glimpse model is implemented as a standard sequence-to-sequence model with attention, where the decoder has a fixed length K. During training, we split the target sequence into non-overlapping, contiguous segments (*glimpses*) with fixed length K, starting from the beginning. We then train on each of these glimpses, one at a time on the decoder, while putting all target-side symbols before the glimpse on the encoder. For example, if a sequence $\mathbf{y}$ is split into two glimpses $\mathbf{y}_1$ and $\mathbf{y}_2$, each with length K ($\mathbf{y}_2$ may be shorter than K), then we will train the model with two examples, $(\mathbf{x} \rightarrow \mathbf{y}_1)$, and $(\mathbf{x}, \mathbf{y}_1 \rightarrow \mathbf{y}_2)$. Each time the concatenated sequence on the left of the arrow is put on the encoder and the sequence on the right is put on the decoder. Figure 1(b) illustrates the training of $(\mathbf{x}, \mathbf{y}_1 \rightarrow \mathbf{y}_2)$ when $K = 3$. In our implementation, we always put the source-side end-of-sequence token at the end of the whole encoder sequence, and we split the glimpses according to the decoder time steps. For example, if the sequence $\mathbf{y}$ is $y_0, y_1, y_2, ..., y_{10}$, and $K = 3$, the first example will have y_0, y_1, y_2 on the input layer of the decoder, and y_1, y_2, y_3 on the output layer of the decoder. The second example has y_3, y_4, y_5 as input of the decoder and y_4, y_5, y_6 as the output of the decoder, and so on. In our experiments, we use $K = 10$.

While decoding each glimpse, the decoder therefore attends to both the source sequence and the part of the target sequence that precedes the glimpse, thereby benefiting from the GNMT encoder's bidirectional RNN. Through generalization, the decoder should learn to decode a glimpse of length K in any arbitrary position of the target sequence (which we will exploit in our decoding technique discussed in Section 4). One drawback of this model, however, is that the context inputs to the attention mechanism only include the words that have been generated so far in this glimpse, rather than the words from the full target side. The workaround that we use is to simply connect the last hidden state of the GNMT-encoder to the initial hidden state of the decoder[1], thereby giving the decoder access to all previous symbols regardless of the starting position of the glimpse.

4 Stochastic Decoding with Segment-by-Segment Reranking

We now turn our attention from training to inference (decoding). Our strategy is to perform reranking with a normalized score at the segment level, where we generate the candidate segments using a trained glimpse-model and using a stochastic beam search procedure, which we discuss next. The full decoding algorithm proceeds segment by segment.

The standard beam search algorithm generates symbols step-by-step by keeping a set of the B highest-scoring beams generated so far at each step[2]. The algorithm adds all possible single-token extensions to every existing beam, and then selects

[1] This is the default in standard seq2seq models, but not in the GNMT model.

[2] Beams are also called ‘hypotheses’, and B is referred to as the ‘beam width’.

the top B beams. In our stochastic beam search algorithm, we replace this deterministic top-B selection by a stochastic sampling operation in order to encourage variation. Further, to discourage a single beam from dominating the search and decreasing the final response diversity, we perform a two-step sampling procedure: 1) For each single-token extension of an individual beam we don't enumerate all possibilities, but instead sample a fixed number of D candidate tokens to be added to the beam. This yields a total of $B \times D$ beams, each with one additional symbol. 2) We then compute the accumulated conditional log-probabilities for each beam (normalized across all $B \times D$ beams), and treat these as the logits for sub-sampling B beams for the next step. We repeat this procedure until we reach the desired segment-length H, or until a segment ends with the end-of-sequence token.

For a given source sequence, we can use this stochastic beam search algorithm to generate B candidate H-length segments as the beginning of the target sequence. We then perform a reranking step (described below), and keep one of these. The concatenation of the source and the first target segment is then used as the input for generating the next B candidate segments. The algorithm continues until the segment selected ends with an end-of-sequence token.

This algorithm behaves similarly to standard beam search when the categorical distribution used during the process is sharp ('peaked'), since the samples are likely to be the top categories (words) . However, when the distribution is smooth, many of the choices are likely. In conversation response generation we are dealing with a conditional probability model with high entropy, so this is what often happens in practice.

For the reranking, we normalize the scores using random prompts. In particular, suppose $\mathbf{y}_k = y_1, ..., y_{k-1}$ is a candidate segment, and $(\mathbf{x}, \mathbf{y}_{1:k-1})$ is the input to the stochastic beam search. The normalized score is then computed as follows:

$$S\left(\mathbf{y}_k|\mathbf{x}, \mathbf{y}_{1:k-1}\right) = \frac{P\left(\mathbf{y}_k|\mathbf{x}, \mathbf{y}_{1:k-1}\right)}{\sum_{\mathbf{x}' \in \Phi} P\left(\mathbf{y}_k|\mathbf{x}', \mathbf{y}_{1:k-1}\right)} \tag{3}$$

In this equation, the set Φ is a collection of randomly sampled source sequences (prompts). In our experiments, we randomly select Q prompts from the context-free evaluation set (introduced in the Experiments section).

It is worth noting that when Φ is an unbiased sample from $P(\mathbf{x})$, the summation in the denominator is a Monte-Carlo approximation of $P(\mathbf{y}_k|\mathbf{y}_{1:k-1})$. In the case of reranking whole target sequences $\mathbf{y}$, this becomes the marginal $P(\mathbf{y})$, which corresponds to the same diversity-promoting objective used in (Li et al., 2015). However, we found that our approximation works better in terms of N-choose-1 accuracy (see Section 5.2), which suggests that its value may be closer to the true conditional probability.

In our experiments, we set number of random prompts Q to 15, segment length H to 10, number of beams B to 2, and samples per beam D to 10. We select a small value for B, since we find that larger values makes the algorithm behave more like standard beam search.

5 Experimental Results

In this section we present experimental results for evaluating the target-glimpse model and the stochastic decoding method that we presented. We train the model using the Google neural machine translation model (GNMT, (Wu et al., 2016)), on a data set that combines multiple sources mined from the Web:

1. The full Reddit data[3] that contains 1.7 billion messages (221 million conversations).

2. The 2009 Open Subtitles data (0.5 million conversations, (Tiedemann, 2009)).

3. The Stack Exchange data (0.8 million conversations).

4. Dialogue-like texts that we recognized and extracted from the web (17 million conversations).

For all these data sets, we extract pairs of messages where one can be considered as a response to the other. For example, in the Reddit data set, the messages belonging to the same post are organized as a tree. A child node is a message that replies to its parent. This may not necessarily be true as people may be replying to other messages that are also visually close. However, for our current single-turn experiments, we treat these as a single exchange.

[3] Download links are at https://redd.it/3bxlg7

In this setting, the GNMT model trained on prompt-to-response pairs works surprisingly well without modification when generating short responses with beam search. Similar to previous work on neural conversation models, we find that the generated responses are almost always grammatical, and sometimes even interesting. They are also usually on topic. In addition, we found that even greedy sampling from the 8-layer GNMT model produces grammatical responses most of the time, although these responses are more likely to be semantically-broken than responses generated using standard beam search. We would like to leverage the benefits of greedy sampling, because the induced variation generates more surprises and may potentially help improve user-engagement, and we found that our proposed segment-based beam sampling procedure accomplishes this to some extent.

5.1 Evaluation Metric

It is difficult to come up with an objective evaluation metric for conversation response generation that can be computed automatically. The conditional distribution $P(\mathbf{y}|\mathbf{x})$ is supposed to have high entropy in order to be interesting (many possible valid responses to a given prompt). Therefore BLEU scores used in translation are not a good fit (also see (Liu et al., 2016)). Other than looking at the evaluation set perplexity, we use two metrics, the **N-choose-1 accuracy** and **5-scale side-by-side human evaluation**. In the N-choose-K metric, we use the model as a retriever. Given a prompt, we ask the model to rank N candidate responses, where one is the ground truth and the other $N-1$ are random responses from the same data set. We then calculate the N-choose-K accuracy as the proportion of trials where the true response is in the top K. The prompts used for evaluation are selected randomly from the same data set. This metric isn't necessarily correlated well with the true response quality, but provides a useful first diagnostic for faster experimental iteration. It takes about a day to train a small model on a single GPU that reaches 2-choose-1 accuracies of around 70% or 80%, but it is much harder to make progress on the 50-choose-1 accuracy. As a reference, human performance on the 10-choose-1 task is around 45% accuracy.

In the 5-scale human evaluation, we use a collection of 200 context-free prompts[4]. These prompts are collected from the following sources, and filtered to prompts that are context-free (*i.e.* do not depend on previous turns in the conversation), general enough, and by eliminating near duplicates:

1. The questions and statements that users asked an internal testing bot.
2. The Fisher corpus (David et al., 2004).
3. User inputs to the Jabberwacky chatbot[5].

These can be either generic or specific. Some example prompts from this collection are shown in Table 1. These prompts are open-domain (not about any specific topic), and include a wide range of topics. Many require some creativity for answering, such as *"Tell me a story about a bear."* Our evaluation set is therefore not from the same distribution as our training set. However, since our goal is to produce good general conversation responses, we found it to be a good general purpose evaluation set.

The evaluation itself is done by human raters. They are well-trained for the purpose of ensuring rating quality, and they are native English speakers. The A 5-scale rating is produced for each prompt-response pair: *Excellent*, *Good*, *Acceptable*, *Mediocre*, and *Bad*. For example, the instructions for rating *Excellent* is "On topic, interesting, shows understanding, moves the conversation forward. It answers the question." The instruction for *Acceptable* is "On topic but with flaws that make it seem like it didnt come from a human. It implies an answer." The instruction for *Bad* is "A completely off-topic statement or question, nonsensical, or grammatically broken. It does not provide an answer."

In our experiments, we perform the evaluations side-by-side, each time using responses generated from two methods. Every prompt-response pair is rated by three raters. We rate 200 pairs in total for every method, garnering 600 ratings overall. After the evaluation, we report aggregated results from each method individually.

5.2 Motivating Experiments

To see whether generating long responses is indeed a challenging problem, we trained the plain

[4] This list will be released to the community.

[5] http://www.jabberwacky.com/

seq2seq with the GNMT model where the encoder holds the source sequence and the decoder holds the target sequence. We experimented with the standard beam search and the beam search with length normalization $\alpha = 0.8$ similar to (Wu et al., 2016). With this length normalization the generated responses are indeed longer. However, they are more often semantically incoherent. It produces *"I have no idea what you are talking about."* more often, similarly observed in (Li et al., 2016). The human evaluation results are summarized in Figure 2(b). Methods that generate longer responses have more *Bad* and less *Excellent* / *Good* ratings.

We also performed the N-choose-1 evaluation on the baseline model using different normalization schemes. The results are shown in Table 2(a). *No Normalization* means that we use $P(\mathbf{y}|\mathbf{x})$ for scoring, *Normalize by Marginal* uses $P(\mathbf{y}|\mathbf{x})/P(\mathbf{y})$, as suggested in (Li et al., 2015), and *Normalize by Random Prompts* is our scoring objective described in Section 4. The significant boost when using both normalization schemes indicates that the conditional log probability predicted by the model may be biased towards the language model probability of $P(\mathbf{y})$. After adding the normalization, the score may be closer to the true conditional log probability.

Overall, this *reranking* evaluation indicates that our heuristic is preferred to scoring using the marginal. However, it is unfortunately hard to directly make use of this score during beam search decoding (*i.e., generation*), since the resulting sequences are usually ungrammatical, as also observed by (Li et al., 2015). This is the motivation for using a segment-by-segment reranking procedure, as described in Section 4.

5.3 Large-Scale Experiments

For our large-scale experiments, we train our target-glimpse model on the full combined data set. Figure 2(d) shows the training progress curve. In this figure, we also include the curve for $K = 1$, that is, the glimpse model with decoder-length 1. It is clear enough that this model progresses much slower, so we terminated it early. However, it is surprising that the glimpse model with $K = 10$ progresses faster than the baseline model with only source-side attention, because the model is trained on examples with decoder-length fixed at 10, while the average response length is 38 in our data set. This means it takes on average 3.8x training steps for the glimpse model to train on the same number of raw training-pairs as the baseline model. Despite this, the faster progress indicates that target-side attention indeed helps the model generalize better.

The human evaluation results shown in Figure 2 compare our proposed method with the baseline seq2seq model. For this, we trained a length-10 target-glimpse model and decoded with stochastic beam-search using segment-by-segment reranking. In our experiments, we were unable to generate better long, coherent responses using the whole-sequence level reranking method from (Li et al., 2015) compared to using standard beam search with length-normalization[6]. We therefore choose the latter as our baseline, because it is the only method which generates responses that are long enough that we can compare to.

Figure 2 shows that our proposed method generates more long responses overall. One third of all responses are longer than 100 characters, while the baseline model produces only a negligible fraction. Although we do not employ any length-promoting objectives in our method, length-normalization is used for the baseline. For responses generated by our method, the proportion of *Acceptable* and *Excellent* responses remains constant or even increases as the responses grow longer. Conversely, human ratings decline sharply with length for the baseline model.

The percentage of test cases with major agreement is high for both methods. We consider a test to have major agreement if two ratings out of the three are the same. For the baseline method, 80% of the responses have major agreements, and for our method it is 70%.

However, shorter responses have a much smaller search space, and we find that standard beam search tends to generate better ("safer") short responses. To maximize cumulative response quality, we therefore implemented a back-off strategy that combines the strengths of the two methods. We choose to fallback to the baseline model without length normalization when the latter produces a response shorter than 40 characters, otherwise we use the response from our method. This corresponds to the white histogram in Figure 2(b). Compared to the other methods in the fig-

[6] This is because the method reranks the responses in the N-best list resulting from the beam search, which tend to be short with not much variation to begin with.

ure, the combined strategy results in more ratings of *Excellent*, *Good*, *Acceptable*, and *Mediocre*, and fewer *Bad* ratings. With this strategy, among the responses generated for the same 200 prompts, 133 were from the standard beam search and 67 were from our model. Out of the 67 long responses, two thirds were longer than 60 characters and half were longer than 75 characters. To compare the combined model's performance with the baseline, we generated responses from both models using the same 200 prompts. For 20 of the response pairs, human raters had no preference, but for the remaining 180, human raters preferred the combined model's response in 103 cases and the baseline's in only 77, indicating a significant win.

6 Conclusion

The research of building end-to-end systems that can engage in general-purpose conversation is still in its infancy. More significant progress is expected to be made with more advanced neural architectures. However, our results reported in this paper show that minimal modeling change and a slightly more advanced decoding technique, combined with training over very large data sets, can still lead to noticeable improvements in the quality of responses generated using neural conversation models. Overall, we found using fixed-lengths in the decoder to make it easier to train on large data sets, as well as to allow us to improve the diversity and coherence of the generated responses earlier during generation, when it has most impact. While the focus of this work has been on conversation modeling, we expect some of these results to carry over to other sequence-to-sequence settings, such as machine translation or image-captioning.

Acknowledgments

We would like to thank Quoc Le, Oriol Vinyals and Jakob Uszkoreit for many helpful discussions, and Scott Benson, Fuchun Peng for collecting the context-free prompt set, and Amin Ahmad for setting up the human evaluation, and Rami Eid, Daniel Cer for collecting training data sets, and Yonghui Wu, Zhifeng Chen, Mike Schuster for help on training the GNMT model.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. http://tensorflow.org/.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Kyunghyun Cho, Bart Van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734.

Christopher Cieri David, David Miller, and Kevin Walker. 2004. The fisher corpus: a resource for the next generations of speech-to-text. In *in Proceedings 4th International Conference on Language Resources and Evaluation*. pages 69–71.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055* .

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *CoRR* abs/1606.01541.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*. ACL, Austin, Texas, page 21222132. https://aclweb.org/anthology/D16-1230.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364* .

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714* .

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pages 3104–3112.

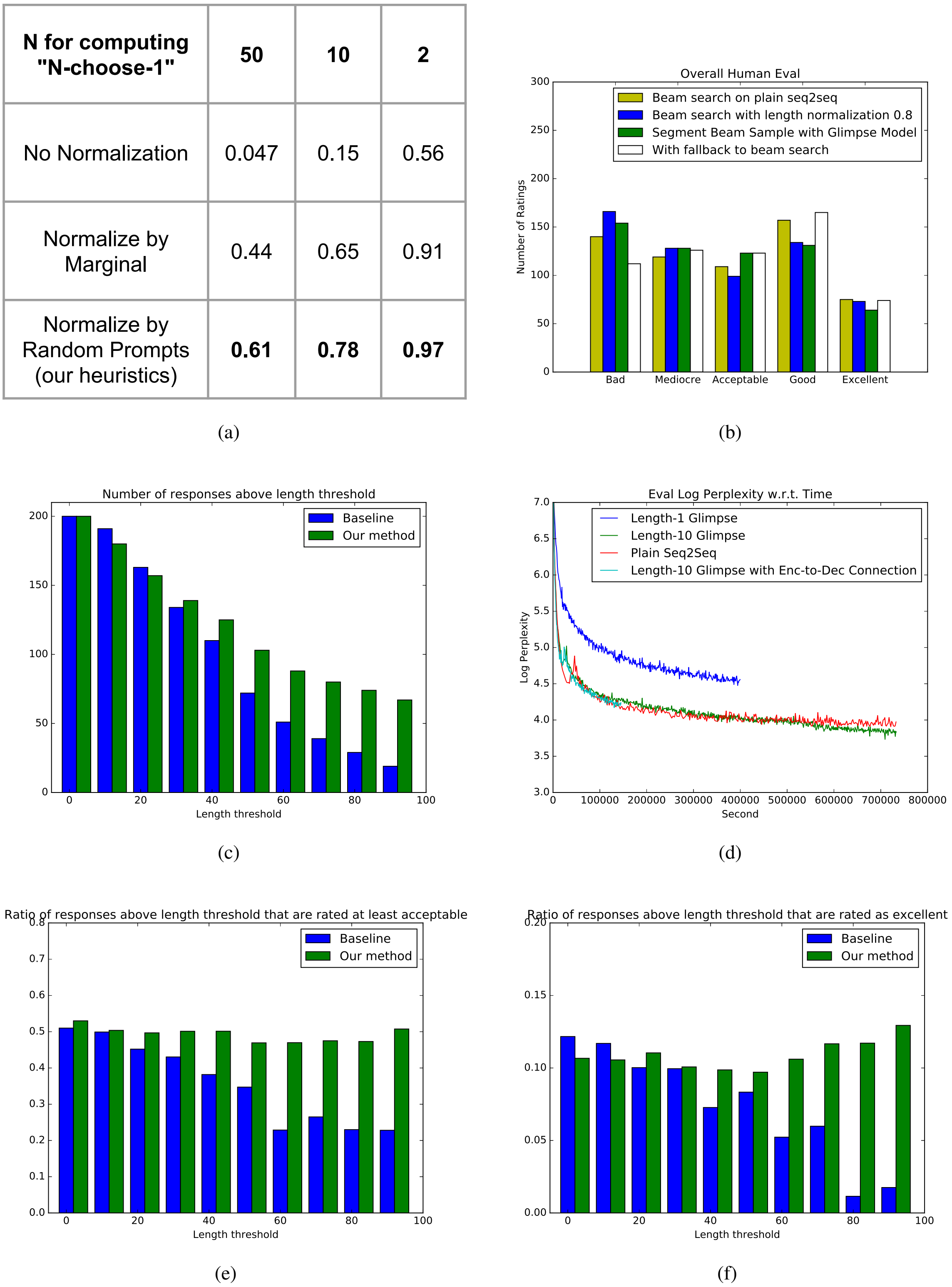
Jörg Tiedemann. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing (vol V)*, John Benjamins, Amsterdam/Philadelphia, pages 237–248.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869* .

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-hao Su, David Vandyke, and Steve J. Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *CoRR* abs/1508.01755.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. Conditional generation and snapshot learning in neural dialogue systems. In *EMNLP*. ACL, Austin, Texas, pages 2153–2162. https://aclweb.org/anthology/D16-1233.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation .



N for computing "N-choose-1"	50	10	2
No Normalization	0.047	0.15	0.56
Normalize by Marginal	0.44	0.65	0.91
Normalize by Random Prompts (our heuristics)	**0.61**	**0.78**	**0.97**

(a)

(b)

(c)

(d)

(e)

(f)

Figure 2: (a) N-choose-1 evaluation on the baseline model. (d) Training progress of different models on the full combined data set. *Length-1* and *Length-10* are the target-glimpse models we propose, and *Plain Seq2seq* is the baseline model we described. (b)(c)(e)(f): Human evaluation results on the conversation data. (b) The histogram of 5 ratings per method. (c) The length thresholds (horizontal axis) and the number of responses generated that are above the length threshold (vertical axis); (e) The proportion of responses above the length-threshold that are judged at least *Acceptable*; (f) The proportion of responses above the length-threshold that are judged as *Excellent*. The length thresholds are all measured in number of characters.