Multi-Turn Beam Search for Neural Dialogue Modeling

Ilia Kulikov*  
New York University

Jason Lee*  
New York University

Kyunghyun Cho  
New York University  
Facebook AI Research  
CIFAR Azrieli Global Scholar

Abstract

In neural dialogue modeling, a neural network is trained to predict the next utterance, and at inference time, an approximate decoding algorithm is used to generate next utterances given previous ones. While this autoregressive framework allows us to model the whole conversation during training, inference is highly suboptimal, as a wrong utterance can affect future utterances. While beam search yields better results than greedy search does, we argue that it is still greedy in the context of the entire conversation, in that it does not consider future utterances. We propose a novel approach for conversation-level inference by explicitly modeling the dialogue partner and running beam search across multiple conversation turns. Given a set of candidates for next utterance, we unroll the conversation for a number of turns and identify the candidate utterance in the initial hypothesis set that gives rise to the most likely sequence of future utterances. We empirically validate our approach by conducting human evaluation using the Persona-Chat dataset [Zhang et al., 2018], and find that our multi-turn beam search generates significantly better dialogue responses. We propose three approximations to the partner model, and observe that more informed partner models give better performance.

1 Introduction

The success of sequence-to-sequence learning [Sutskever et al., 2014; Cho et al., 2014] has sparked interest in applying neural autoregressive models to dialogue modeling [Vinyals and Le, 2015]. In this paradigm, the problem of dialogue modeling has largely been treated as the problem of next utterance prediction, in which the goal is to build a neural sequence model that produces a distribution over next utterances, given previous utterances and any relevant context.

We make an observation that while maximizing log-probability of the next utterance is equivalent to maximizing log-probability of the whole conversation during training, this does not hold during inference with approximate decoding: consecutively choosing the most likely next utterance (utterance-level inference) may not lead to the most likely conversation overall (conversation-level inference). When simply choosing the most likely next utterance, its impact on the future utterances is not taken into account, therefore a suboptimal choice can be made.

We propose a new decoding algorithm, called “multi-turn beam search”, to approximately solve conversation-level inference. This algorithm rolls out multiple future conversation trajectories from each candidate utterance in the hypothesis set from initial utterance-level inference (e.g. beam search) and selects the one with the highest conversation-level log-probability. To this end, we introduce a partner model that approximates the unknown behaviour of the partner. We explore three possibilities of the partner model: mindless partner model, egocentric model and transparent partner model.

* these authors contributed equally
We evaluate the proposed search strategy by having a neural dialogue model engage in a full conversation with human annotators using the ParlAI framework (Miller et al., 2017). We empirically observe that annotators rate the multi-turn approach significantly higher than conventional beam search, and increasing the number of lookahead steps results in better performance. Among the three proposed partner models, the egocentric model and the transparent model result in better conversations than the mindless partner does. This implies that having an informative model of the partner, even if incorrect, helps generate better dialogue responses by narrowing down the space of potential future utterances. Also, our approach can be used with any utterance-level search algorithm, and we verify that its performance is not sensitive to this choice by comparing vanilla beam search and iterative beam search (Kulikov et al., 2018).

2 Neural Dialogue Modeling

Neural dialogue modeling is a framework in which a neural network is used to model a full conversation between two speakers (Vinyals and Le, 2015). A conversation consists of a sequence of utterances alternating between two speakers, self and partner, a set of context information of which some parts are only available to each speaker. We represent the conversation as
\[ C = (X_1^1, X_1^p, \ldots, X_M^1, X_M^p, C^s, C^p), \]
where \( C^s \) and \( C^p \) are static context information, fixed throughout the conversation and only visible to the speakers \( s \) and \( p \), respectively. \( X_m^s \) and \( X_m^p \) are the utterances of the speakers \( s \) and \( p \) at the \( m \)-th turn.

We factorize the distribution over these conversations as:
\[
P((X_1^s, X_1^p, \ldots, X_M^s, X_M^p) | C^s, C^p) = \prod_{m=1}^{M} p(X_m^s | X_{<m}^s, X_m^p, C^s, C^p) \]
\[
= \prod_{m=1}^{M} p_s(X_m^s | X_{<m}^s, X_m^p, C^s) \times p_p(X_m^p | X_{<m}^p, X_m^p, C^p),
\]
(1)
where we assume that the self always initiates the conversation and that the two speakers alternate. This factorization allows us to utilize two separate neural networks as utterance models of \( s \) and \( p \), respectively. In other words, each speaker uses its own context and previous utterances to generate the next utterance.

Each speaker model is further factorized into a series of next token predictions:
\[
p_i(X_i | X_{<i}, C^i) = \prod_{t=1}^{T} p_i(x_{m,t} | x_{m,t-1}, X_{<m}, X_{<m}^i, C^i),
\]
where \( x_{m,t}^i \) is the \( t \)-th token in \( X_{m}^i, \) and \( i \in \{ s, p \} \). \( i \) is \( p \) if \( i \) is \( s \) and otherwise \( s \).

2.1 Learning

Given a training set of conversations, we can train these neural networks to maximize its log-likelihood which is the sum of the per-conversation log-probabilities \( \mathcal{L}(C) \):
\[
\mathcal{L}(C) = \sum_{m=1}^{M} \log p(X_m^s | X_{<m}^s, X_m^p, C^s) + \log p(X_m^p | X_{<m}^p, X_m^p, C^p).
\]
(2)
This formulation implies that learning is equivalent to training a neural network to predict a correct response \( X_m^i \) at the \( m \)-th turn given the history of recorded responses from both speakers and the corresponding context \( C^i \), where \( i \in \{ s, p \} \). Therefore, next utterance prediction is equivalent to full conversation modeling in terms of learning (maximizing log-likelihood).

2.2 Inference

Although the autoregressive factorization reduces dialogue modeling to next utterance prediction for learning, we show in this section that this is not the case for inference where the goal is to find the most appropriate utterance by speaker \( i \) at time \( t \) under a trained neural dialogue model.
**Utterance-Level Inference** When neural dialogue modeling is viewed as next utterance prediction, it is natural to formulate inference as

$$\hat{X}_m^i = \arg \max_{X_m^i} p_i(X_m^i | X_{<m}^i, C^i), \quad (3)$$

where $p_i$ was defined earlier in Eq. (2). It is intractable to solve this problem exactly, and it is common to resort to approximate search algorithms, such as greedy search, beam search and iterative beam search (see, e.g., Cho 2016; Kulikov et al. 2018 for detailed descriptions). We call this inference procedure “utterance-level inference”.

**Conversation-Level Inference** When modeling a full conversation as in Eq. (1), it is necessary to take into account the impact of an utterance on the future utterances. There are two ways to measure this. First, we can maximize the conditional probability of the utterance of speaker $i$ at turn $t$ given the history after considering all possible future utterances.

$$\hat{X}_m^i = \arg \max_{X_m^i} p_i(X_m^i | X_{<m}^i, X_{<m}^i, C^i) \times \left[ \sum_{X_{>m}} \prod_{m'=m+1}^M p(X_{m'}, X_{m'} | X_{<m'}, X_{<m'}, C^s, C^p) \right]. \quad (4)$$

We call this conservative conversation-level inference. When exact search is used, utterance-level inference implicitly marginalizes out future utterances, and is equivalent to conservative conversation-level inference. With approximate decoding, utterance-level and conversation-level inference are no longer equivalent due to the error being introduced by the approximation.

However, considering all future utterances is intractable and potentially unnecessary in conversation-level inference. Therefore, we propose only taking the most likely future utterances instead of marginalization:

$$\hat{X}_m^i = \arg \max_{X_m^i} p_i(X_m^i | X_{<m}^i, X_{<m}^i, C^i) \times \left[ \max_{X_{>m}} \prod_{m'=m+1}^M p(X_{m'}, X_{m'} | X_{<m'}, X_{<m'}, C^s, C^p) \right]. \quad (5)$$

In this work, we focus on the latter option and refer to it as optimistic “conversation-level inference”.

**Mismatch** When approximate decoding is used, the solution to utterance-level inference in Eq. (3) does not generally coincide with that to either conversation-level inference strategy in Eqs. (4)–(5). The cause of this mismatch is akin to the suboptimality of greedy decoding in utterance-level inference. That is, the choice at turn $t$ influences the rest of the conversation. A good choice of utterance according to Eq. (3) may lead to unlikely future utterances.

### 3 Multi-Turn Beam Search

The goal of this work is to empirically investigate the mismatch between the utterance-level and conversation-level inference strategies. To do so, we propose an approximate algorithm for solving the optimistic conversation-level inference problem in Eq. (5). We call this algorithm “multi-turn beam search”.

The context in which a conversation is conducted consists of $C^s$ and $C^p$ which are only visible to speakers $s$ and $p$, respectively. This implies that we cannot compute the log-probability of the future utterances (the second term in Eq. (5)) exactly, because we cannot assume access to the conversation partner, except for observing her utterances. We therefore assume access to an approximate model of the “partner”, $p_p$, in addition to the model of it’s “self”, $p_s$.

Consider the $m$-th turn in an ongoing conversation. We first compute a set of $K$ likely utterances from the self model $p_s$ using a variant of beam search:

$$\mathcal{H}_0 = \left\{ (\hat{X}_0^s, S_{0,1}), \ldots, (\hat{X}_0^K, S_{0,K}) \right\},$$

where

$$S_{0,k} = \log p_s(\hat{X}_0^s | X_0^s, X_{<m}^p, C^s).$$
Selecting the candidate that maximizes $S_{0,k}$ corresponds to utterance-level inference. We instead run conversation-level inference using a partner model, $p_p$. Given the $k$-th candidate in $H_0$, we use beam search (or any utterance-level inference algorithm) to generate $K$ candidate responses in the partner model:

$$
\hat{H}_{0,k} = \left\{ (\hat{X}^p_{0,k,1}, S_{0,k,1}), \ldots, (\hat{X}^p_{0,k,K}, S_{0,k,K}) \right\},
$$

where

$$
S_{0,k,k'} = S_{0,k} + \log p_p(\hat{X}^p_{0,k,k'}|X^s_{\leq m}, X^p_{\leq m}, C^p).
$$

This procedure leads to $K \times K$ candidate utterance sequences of two utterances each. We select top-$K$ utterance sequences among these to form the next candidate set:

$$
\mathcal{H}_1 = \left\{ (\hat{X}^s_{1,1}, S_{1,1}), \ldots, (\hat{X}^s_{1,K}, S_{1,K}) \right\},
$$

where

$$
\hat{X}^s_{1,k} = [\hat{X}^p_{0,k',k''}, \hat{X}^p_{0,k',k''}], \quad S_{1,k} = S_{0,k',k''},
$$

and $(k', k'')$ is the $k$-th best candidate from the new candidate sequences.

We iterate this procedure up to $L$ iterations (look-ahead steps) to obtain $K$ candidate (future) utterance sequences and pick the one with the highest overall score $S_{L,k}$. Then, the first utterance in the best utterance sequence is used as the new utterance.

### 3.1 Properties

The proposed approach has two properties that are worth discussing. First, it works at the conversation level, meaning that we can plug in any utterance-level inference algorithm, as long as it returns more than one candidate. This choice certainly influences the conversation-level inference, and we investigate its influence later with two different utterance-level algorithms. Second, the proposed algorithm cannot generate an utterance that is outside the initial candidates $H_0$ from the utterance-level inference algorithm. Although this constraint can be sidestepped by extending the proposed algorithm to make a backward pass, we leave this as future work.

**Computational Complexity** The computational complexity of the proposed approach grows linearly with respect to the number of look-ahead turns $L$. Each look-ahead turn incurs $O(KT \log K + K^2 \log K)$, as we run beam search $O(T \log K)$ over up to $T$ tokens $K$ times (for each candidate utterance sequence) and select top-$K$ utterance sequences out of $K^2$ candidates. This is simplified to $O(KT \log K)$, as $T \gg K$ often. This procedure is run $L$ times, leading to the overall computational complexity of $O(LKT \log K)$. Compared to the conventional greedy approach, the proposed algorithm introduces the multiplicative factor of $LK$. Since $L$ and $K$ are both small integers, we do not expect too much computational overhead in practice.

### 3.2 Partner Models

The most notable feature of the proposed algorithm is explicitly modeling the dialogue partner $p_p(X^p_m|X^s_{\leq m}, X^p_{\leq m}, C^p)$. The main difficulty lies in the fact that at test time, the neural dialogue model converses with an unknown partner with unknown context $C^p$. We address this issue by building approximations to the true partner model and its true context $C^p$. We explore three options for the partner model, although we anticipate other approaches to be developed in the future.

**Mindless Partner** The most naive solution to this issue is to train a separate partner model $p_{\text{less}}$ that does not depend on the context $C^p$, i.e.,

$$
p_p(X^p_m|X^s_{\leq m}, X^p_{\leq m}, C^p) \approx p_{\text{less}}(X^p_m|X^s_{\leq m}, X^p_{\leq m}). \tag{6}
$$

This “mindless partner model” can be trained to predict the next utterance based solely on the previous utterances without having access to the context. It is also possible to view this approach as marginalizing out the effect of the context on utterance prediction.
Another approach is to assume that the partner is identical to the self model.

\[
p_p(X_p^m | X_s^{\leq m}, X_p^{< m}, C_p) \approx p_s(X_p^m | X_s^{\leq m}, X_p^{< m}, C_s).
\]  

This “egocentric model” of the partner makes a strong assumption that the partner shares the mental states, beliefs and intentions of the self model. Although these assumptions are likely to be wrong, it nevertheless provides a useful signal as to the candidates that would lead to a more likely sequence of “future” utterances.

**Transparent Partner**  
In addition to the two partner models above, we explore a “transparent partner” whose real context \( C_p \) is fully exposed. In other words, we condition the self model \( p_s \) on \( C_p \) to get an approximation to the true partner model:

\[
p_p(X_p^m | X_s^{\leq m}, X_p^{< m}, C_p) \approx p_s(X_p^m | X_s^{\leq m}, X_p^{< m}, C_p).
\]

### 4 Experimental Setup

We focus on three aspects of the proposed algorithm. First, we test whether the proposed multi-turn approach outperforms the conventional approach, and investigate the effect of increasing the number of look-ahead steps in the conversation-level inference. Second, we vary our approximation to the partner model, between the mindless, egocentric and transparent partner. This allows us to understand the influence of our assumption on the partner’s behaviour. Last, we investigate the sensitivity of the proposed conversation-level inference to the choice of the utterance-level inference algorithm.

#### 4.1 Dataset: Persona-Chat

We use Persona-Chat \cite{Zhang2018} to train a neural dialogue model. The dataset contains dialogues between pairs of annotators who are each assigned a randomly chosen persona from a set of 1,155. Concretely, annotators are shown 4-5 lines of description of the role they are asked to play in the dialogue, e.g. “I have two dogs” or “I like taking trips to Mexico”. The training set consists of 9,907 dialogues where pairs of annotators engage in a conversation assuming their randomly assigned personas, and a validation set of 1,000 dialogues. The test set has not been released. Each dialogue is tokenized into words, resulting in a training vocabulary of 18,760 unique tokens. Each dialogue in the training data is 6.84 turns long on average. See \cite{Zhang2018} for more details.

#### 4.2 Models and Learning

We closely follow \cite{Bahdanau2015} in building an attention-based neural autoregressive sequence model for each speaker model. The encoder has two bidirectional layers of 512 LSTM \cite{Hochreiter1997} units, and the decoder has two layers of 512 LSTM units each. We use global general attention as described by \cite{Luong2015}. We share the embeddings between the encoder and the decoder, which are initialized as 300-dimensional pretrained GloVe vectors \cite{Pennington2014}. We update word embedding weights during the training.

**A Self Model** is trained by conditioning on the self model’s persona \( C_s \). During inference, it is used as the self model \( p_s \). **A Mindless Partner Model** is separately trained without conditioning on any persona. During inference, it is used to approximate \( p_p \). **An Egocentric Partner Model** is using the self model at inference time, while conditioning on the model’s persona \( C_s \) to approximate the partner’s distribution \( p_p \). **A Transparent Partner Model** is using the self model at inference time, while conditioning on the true partner persona \( C_p \).

**Learning**  
We use Adam \cite{Kingma2014} with the initial learning rate set to 0.001. We apply dropout \cite{Srivastava2014} between the LSTM layers with rate of 0.5. We train the neural dialogue model until it early-stops on the validation set.

#### 4.3 Evaluation

**Human Evaluation**  
We use ParlAI \cite{Miller2017} which provides seamless integration with Amazon Mechanical Turk for human evaluation. A human annotator is paired with a model with a
Table 1: NLL and human evaluation score of each inference strategy. Steps: the number of lookahead steps. Width: beam width. NLL: average negative log likelihood per conversation. Score: human judgment score (in a scale of 0–3) after calibration (with standard deviation). For multi-turn approaches, we used an egocentric partner model.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Width</th>
<th>NLL</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>3.57</td>
<td>1.67 ± 0.16</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>3.52</td>
<td>1.50 ± 0.18</td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>3.37</td>
<td>1.45 ± 0.18</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>3.36</td>
<td>1.80 ± 0.17</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>3.19</td>
<td>1.81 ± 0.18</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>3.20</td>
<td>1.98 ± 0.19</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>3.96</td>
<td>1.86 ± 0.19</td>
</tr>
</tbody>
</table>

Human 2.66 ± 0.22

Table 2: Candidate selection statistics. Steps: the number of lookahead steps. Rate: the average frequency with which the multi-turn approach selects a different candidate from the conventional beam search. Rank: the average rank of the candidate chosen by the multi-turn approach in the initial beam hypothesis set. Gap: the average drop in log-probability from the top to the second-best candidates in the initial set of hypotheses.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Beam</th>
<th>Iterbeam</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rate</td>
<td>Rank Gap</td>
</tr>
<tr>
<td>1</td>
<td>0.37</td>
<td>0.77</td>
</tr>
<tr>
<td>2</td>
<td>0.34</td>
<td>0.66</td>
</tr>
<tr>
<td>4</td>
<td>0.42</td>
<td>0.81</td>
</tr>
<tr>
<td>8</td>
<td>0.47</td>
<td>1.18</td>
</tr>
</tbody>
</table>

As running grid search on all hyperparameters would have been costly, we carefully selected 29 hyperparameter combinations including the vanilla beam search as baseline.
Table 3: Human evaluation scores with respect to different approximation to the partner model. The egocentric model performs on par with the transparent model, while the mindless model performed worse in case of 2 look-ahead steps.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Mindless</th>
<th>Egocentric</th>
<th>Transparent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.67</td>
<td>1.67</td>
<td>1.67</td>
</tr>
<tr>
<td>1</td>
<td>1.83</td>
<td>1.80</td>
<td>1.82</td>
</tr>
<tr>
<td>2</td>
<td>1.51</td>
<td>1.81</td>
<td>1.78</td>
</tr>
<tr>
<td>4</td>
<td>1.87</td>
<td>1.98</td>
<td>1.96</td>
</tr>
<tr>
<td>8</td>
<td>1.72</td>
<td>1.64</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Human evaluation scores with respect to different utterance-level inference algorithms, using the egocentric partner model. Beam: vanilla beam search. Iterbeam: iterative beam search (Kulikov et al., 2018).

<table>
<thead>
<tr>
<th>Steps</th>
<th>Beam</th>
<th>Iterbeam</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.67</td>
<td>1.58</td>
</tr>
<tr>
<td>1</td>
<td>1.80</td>
<td>1.73</td>
</tr>
<tr>
<td>2</td>
<td>1.81</td>
<td>1.62</td>
</tr>
<tr>
<td>4</td>
<td>1.98</td>
<td>1.68</td>
</tr>
<tr>
<td>8</td>
<td>1.86</td>
<td>1.92</td>
</tr>
</tbody>
</table>

Does multi-turn beam search help? In Table 1, we compare results between different numbers of look-ahead turns. We make two key observations. First, simply increasing the beam size does not lead to better responses in terms of human judgment. In fact, we notice that high ranking candidates in a large beam are often generic and short. Second, increasing the number of look-ahead steps leads to significantly better dialogue responses. Negative log-likelihood scores show that conversations obtained using multi-turn beam search has higher probability compared to vanilla beam search, although the score degrades with 8 turns. This is not surprising, as the average number of turns in the training set is only 6.8, therefore the estimate of the conversation log-probability is not reliable. Finally, human-to-human conversation is rated far higher than human-to-model conversations, indicating much room for further improvement in dialogue modeling research.

Are different candidates selected? Table 2 presents statistics of our multi-turn beam search. First, the proposed approach starts to disagree more with the utterance-level inference as the number of lookahead steps increases. The rate and rank of the iterative beam search are lower than those of vanilla beam search. This happens due to the larger average drop in log-probability between first two best candidates. A larger difference on the utterance-level is more difficult to overcome in the future on the conversation-level.

How important is the partner model? We observe that the mindless partner (mindless in Table 3) is the worst performer: its quality drops significantly at two lookahead turns already. On the other hand, the egocentric and transparent models generate increasingly better responses with more lookahead steps, until performance degrades with 8 lookahead steps.

We draw two conclusions. First, even when an incorrect context information is provided to the partner model, it is beneficial to produce less generic utterances using a model with sharp distribution. Second, the transparent partner model gives no improvement over the egocentric model. Hence, we need a more powerful dialogue model to be able to take full advantage of the context information.

How sensitive is multi-turn beam search to utterance-level inference? Table 4 presents the comparison of human evaluation scores between vanilla beam search and iterative beam search. We do not observe any significant difference between the two—performing more lookahead steps with iterative beam search leads to better dialogue responses as well as with vanilla beam search. This shows that our multi-turn approach is robust to the choice of the search algorithm used at the utterance level.

5.2 Qualitative Results

In Figure 1, we show a visualization of multi-turn beam search with one lookahead step. Given the model context and previous utterances (shown in the left box), utterance candidates from the initial beam search are in the middle (sorted with respect to log probability), and the candidates for the next turn are shown on the right (also sorted). The most likely utterance in the initial beam search (“i love music! what kind of music do you like?”) is not a plausible answer to the previous question (“great! i’ll be 101 tomorrow, you like radio program?”). The multi-turn approach selects a more reasonable response given the context (“i do. i like to dance.”) by performing one lookahead step using a partner model, which in turn selects a sensible response (“what kind of music do you like?”).
In this paper, we demonstrate that explicitly modeling the dialogue partner and accounting for future utterances is beneficial for inference in neural dialogue modeling. We motivate this by making an observation that there is a mismatch between next utterance prediction and full conversation modeling in approximate decoding. During training, maximizing the likelihood of the next utterance is equivalent to maximizing the likelihood of the whole conversation. During inference, however, choosing the most likely utterance (or its approximation, such as given by beam search) is likely to be sub-optimal, as it does not account for future utterances.
From our human evaluation where annotators engaged in a full conversation with our models, we draw the following conclusions. First, the proposed multi-turn approach is able to select significantly better responses than regular beam search. Second, the choice of a partner model affects the effectiveness of our approach: an egocentric or a transparent partner model significantly outperforms a mindless partner model. Last, our approach is robust to the choice of the utterance-level inference algorithm.

While the Persona-Chat dataset is the only dataset that provides context information to each speaker, its conversations are rather short. We anticipate more insight to be gained from experimenting with longer conversations in the future. Furthermore, more sophisticated partner models need to be tested in order to exploit the full capabilities of the proposed search algorithm.

References


