Bidirectional Beam Search: Forward-Backward Inference in Neural Sequence Models for Fill-in-the-Blank Image Captioning

Qing Sun
Virginia Tech
sunqing@vt.edu

Stefan Lee
Virginia Tech
steflee@vt.edu

Dhruv Batra
Georgia Tech
dbatra@gatech.edu

Abstract

We develop the first approximate inference algorithm for 1-Best (and M-Best) decoding in bidirectional neural sequence models by extending Beam Search (BS) to reason about both forward and backward time dependencies.

Beam Search (BS) is a widely used approximate inference algorithm for decoding sequences from unidirectional neural sequence models. Interestingly, approximate inference in bidirectional models remains an open problem, despite their significant advantage in modeling information from both the past and future. To enable the use of bidirectional models, we present Bidirectional Beam Search (BiBS), an efficient algorithm for approximate bidirectional inference.

To evaluate our method and as an interesting problem in its own right, we introduce a novel Fill-in-the-Blank Image Captioning task which requires reasoning about both past and future sentence structure to reconstruct sensible image descriptions. We use this task as well as the Visual Madlibs dataset to demonstrate the effectiveness of our approach, consistently outperforming all baseline methods.

1. Introduction

Recurrent Neural Networks (RNNs) and their generalizations (LSTMs, GRUs, *etc.*) have emerged as a popular and effective framework for modeling sequential data across varied domains. The application of these models has led to significantly improved performance on a variety of tasks – speech recognition [1,2], machine translation [3–5], conversation modeling [6], image captioning [7–11], visual question answering (VQA) [12–16], and visual dialog [17, 18].

Broadly speaking, in these applications RNNs are typically used in two distinct roles – (1) as *encoders* that convert sequential data into vectors, and (2) as *decoders* that convert encoded vectors into sequential output. Models for image caption retrieval and VQA (with classification over answers) [12, 13] consist of encoder RNNs but not decoders. Image caption generation models [7] consist of decoder RNNs but not encoders (image encoding is performed

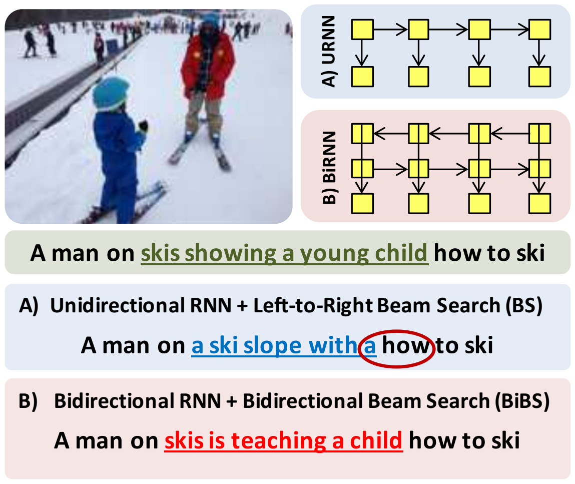

A) URNN
B) BiRNN
A man on skis showing a young child how to ski
A) Unidirectional RNN + Left-to-Right Beam Search (BS)
A man on a ski slope with a how to ski
B) Bidirectional RNN + Bidirectional Beam Search (BiBS)
A man on skis is teaching a child how to ski

Figure 1: We develop a novel Bidirectional Beam Search (BiBS) algorithm for neural sequence models and propose a new fill-in-the-blank image captioning task as a challenging testbed for sequence completion. Unidirectional RNNs fail to reason about both past and future outputs and produce nonsensical outputs for this task – note the jarring "with a how to" transition produced by classical beam search in (A). In contrast, our BiBS algorithm on a Bidirectional RNN produces significantly better completions (B) by considering context on either side of the blank.

via Convolutional Neural Networks). Visual dialog models use encoders to embed dialog history and model state, while using decoders to generate dialog responses. Regardless of the setting, the task of decoding a sequence from an an RNN consists of finding the most likely sequence $Y = (\mathbf{y}_1, ..., \mathbf{y}_T)$ given some input $\mathbf{x}$.

Unidirectional RNNs model this probability by estimating the likelihood of outputting a symbol at time t (say $\mathbf{y}_t$) given the history of previous outputs $(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1})$ by "compressing" the history into a hidden state vector $\mathbf{h}_{t-1}$ such that $\mathrm{P}(\mathbf{y}_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{x}) \simeq \mathrm{P}(\mathbf{y}_t \mid \mathbf{h}_{t-1})$. Since each output symbol is conditioned on all previous outputs, the search space of possible sequences is exponential in the sequence length and exact inference is intractable. As a result, approximate inference algorithm are applied, with the Beam

Search (BS) being the primary workhorse. BS is a greedy heuristic search that maintains the top-B most likely partial-sequences through the search tree, where B is referred to as the *beam width*. At each time step, BS *expands* these B partial sequences to all possible beam extensions and then selects the B highest scoring among the expansions.

In contrast to Unidirectional RNNs, Bidirectional RNNs model both forward (increasing time) and backward (decreasing time) dependencies $\mathrm{P}(\mathbf{y}_t \mid \mathbf{h}_t^f, \mathbf{h}_t^b)$ via two hidden state vectors $\mathbf{h}_t^f$ and $\mathbf{h}_t^b$. This enables Bidirectional RNNs to consider both past and future when predicting an output. Unfortunately, these dependencies also make exact inference in these models more difficult than in Unidirectional RNNs and to the best of our knowledge no efficient approximate algorithms exist. In this paper, we present the first efficient approximate inference algorithm for these models.

As a challenging testbed for our method, we propose a fill-in-the-blank image captioning task. As an example, given the blanked image caption *"A man on ________________ how to ski"* shown in Figure 1, our goal is to generate the missing content *"skis showing a young child"* (or an acceptable paraphrase) to complete the sentence. This task serves a concrete stand-in for a broad class of other similar sequence completion tasks, such as predicting missing sections in a DNA sequence or path planning problems where an agent must hit intermediate flag points.

On the surface, this task perhaps seems easier than generating an entire caption from scratch; there is after all, more information in the input. However, the need to condition on the context when generating the missing symbols is challenging for existing greedy approximate inference algorithms. Figure 1(a) shows a sample decoding from standard 'left-to-right' BS on a Unidirectional RNN. Note the grammatically incorrect "with a how to" transition produced. Similar problems occur at the other boundary for right-to-left models. Simply put, the inability to consider both the future and past contexts in BS leads Unidirectional RNNs to fill the blank with words that clash abruptly with the context around the blank.

Moreover, decoding also poses a computational challenge. Consider the following sentence that we know has only a single word missing: *"The _____ was barreling down the tracks."* Filling in this blank feels simple – we just need to find the best single word in the vocabulary $y_t \in \mathcal{Y}$. However, since all future outputs in a Unidirectional RNN are conditioned on the past, selecting the best word at time t requires evaluating the likelihood *of the entire sequence* once for each possible word $y_t \in \mathcal{Y}$ (similarly for Bidirectional RNNs). This amounts to $T|\mathcal{Y}|$ forward passes through an RNN's basic computational unit *to fill in a single blank optimally*! More generally, for an arbitrarily sized blank covering w words, this number grows exponentially as $T|\mathcal{Y}|^w$ and quickly becomes intractable.

To overcome these shortcomings, we introduce the first approximate inference algorithm for 1-Best (and M-Best) inference in bidirectional neural sequence models (RNNs, LSTMs, GRUs, *etc.*) – Bidirectional Beam Search (BiBS). We show BiBS performs well on fill-in-the-blank tasks, efficiently incorporating both forward and backward time information from Bidirectional RNNs.

To give an algorithmic overview, we begin by decomposing a Bidirectional RNN into two calibrated but independent Unidirectional RNNs (one going forward in time and the other backward). To perform approximate inference with these decomposed models, our method alternatively performs BS on one direction while holding the beams in the opposite direction fixed. The fixed, oppositely-directed beams are used to roughly approximate the conditional probability of all future sequence given the past such that a BS-like update minimizes an approximation of the full joint at each time step. Figure 1(b) shows an example result of our algorithm – "A man on skis is teaching a child how to ski" – which smoothly fits within its context while still describing the image content.

We compare BiBS against natural ablations and baselines for fill-in-the-blank tasks. Our results show that BiBS is an effective and efficient approach for decoding Bidirectional RNNs, consistently outperforming all baselines.

2. Related Work

While Unidirectional RNNs are popular models with widespread adoption [1–6,12,13], Bidirectional RNNs have been utilized in relatively infrequently [19–21] and even more rarely as decoders [22] – we argue due to the lack of efficient inference approaches for these models.

Wang *et al.* [21] used Bidirectional RNNs for image caption generation, but do not perform bidirectional inference, rather simply use BiDirectional RNNs to rescore candidates. Specifically, at inference time they decompose a Bidirectional RNN into two independent Unidirectional RNN, apply standard Beam Search in each direction, and then reranked these two collection of beams based on the max probability of each beam under the forward or backward Unidirectional RNN model. We compare to this method in our experiments and show that joint optimization via Bidirectional Beam Search leads to better sequence completions for our fill-in-the-blank image captioning task.

Most related to our work is that of Burglund *et al.* [22], which studies generating missing data in time series data in an unsupervised setting using Bidirectional RNNs. They propose three probabilistically justified approaches to fill these gaps by drawing samples from the full joint.

Their first model, Generative Stochastic Networks (GSN), resamples the output $\mathbf{y}_t$ at a random time t from the conditional output $\mathrm{P}(\mathbf{y}_t \mid Y_{[1:T]\setminus t})$. For a blank of length w, resampling each output tokens M times requires wMT passes of the RNN. Thus, the cost of producing a sample with the GSN method scales linearly with the size of the gap and requires a full pass of the Bidirectional RNN.Their second approach, NADE, trains a model specifically for filling in the blank – *i.e.* at train time, some inputs are set to a specific 'missing' token to indicate the content that needs to be generated. At inference time, the inputs from the gap are set to this token and sampled from the resulting conditional. Note that this approach is 'trained to fill in gaps' and as such requires training data of this kind. To contrast, this is a new model for filling in gaps, while we propose a new inference algorithm, which can be broadly applied to any generative bidirectional model. Finally, they propose a third sampling approach based on a Unidirectional RNN which draws from the conditional $\mathrm{P}(\mathbf{y}_t \mid Y_{[1:T]\setminus t})$; however, as the model is a left-to-right Unidirectional RNN, this term requires computing the likelihood of the remaining sequence given each possible token at time t. This costly approach requires $w|\mathcal{Y}|MT$ steps of the RNN and is intractable for large vocabularies.

3. Preliminaries: RNNs and Beam Search

We begin by establishing notation, and reviewing RNNs and standard Beam Search for completeness. While our exposition details the classical RNN updates, the techniques developed in this paper are broadly applicable to any recurrent neural architecture (*e.g.* LSTMs [23] or GRUs [24]).

Notation. Let $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ denote an input sequence, where $\mathbf{x}_t$ is an input vector at time t. Similarly, let $Y = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T)$ denote an output sequence, where $\mathbf{y}_t$ is an output vector at time t. To avoid notational clutter, our exposition uses the same length for both input and output sequences (T); however, this is not a restriction in theory or practice. Given integers a, b, we use the notation $Y_{[a:b]}$ to denote the sub-sequence $(\mathbf{y}_a, \mathbf{y}_{a+1}, \ldots, \mathbf{y}_b)$; thus, $Y = Y_{[1:T]}$ by convention. Given discrete variables Y, we generalize the classical maximization notation $\mathrm{argmax}_{Y \in \mathcal{Y}}\ f(Y)$ to find the (unique) top B states with highest $f(Y)$ via the notation $\text{top-B}_{Y \in \mathcal{Y}}\ f(Y)$.

Unidirectional RNN (URNNs) model the probability of $\mathbf{y}_t$ given the history of inputs $\mathbf{x}_1, \ldots, \mathbf{x}_t$ by "compressing" the history into a hidden state vector $\mathbf{h}_t$ such that

$$\mathrm{P}(\mathbf{y}_t \mid X_{[1:t]}) = \phi(W_y \mathbf{h}_t + \mathbf{b}_y) \tag{1a}$$
$$\mathbf{h}_t = \tanh(W_x \mathbf{x}_t + W_h \mathbf{h}_{t-1} + \mathbf{b}_h) \tag{1b}$$

where $W_x, W_h, W_y, \mathbf{b}_h$, and $\mathbf{b}_y$ are learned parameters defining the transforms from the input $\mathbf{x}_t$ and hidden state $\mathbf{h}_{t-1}$ to the output $\mathbf{y}_t$ and updated hidden state $\mathbf{h}_t$. In applications with symbol sequences as output (such as image captioning), the nonlinear function ϕ is typically the softmax function which produces a distribution over the output vocabulary $\mathcal{Y}$. An example left-to-right Unidirectional RNN architecture is shown in Figure 2a.

Bidirectional RNNs (BiRNNs) (shown in Figure 2b) model both forward (positive time) and backward (negative time) dependencies via two hidden state vectors – forward $\overrightarrow{\mathbf{h}}_t$ and backward $\overleftarrow{\mathbf{h}}_t$ – each with its own update dynamics and corresponding weights. For a BiRNN, we can write the probability of the token $\mathbf{y}_t$ given the input sequence as

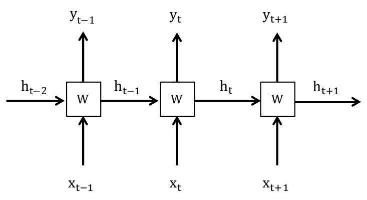
$$\mathrm{P}(\mathbf{y}_t \mid X_{[1:T]}) = \phi(\underbrace{\overrightarrow{W}_y \overrightarrow{\mathbf{h}}_t}_{\text{Forward score}} + \underbrace{\overleftarrow{W}_y \overleftarrow{\mathbf{h}}_t}_{\text{Backward score}} + \mathbf{b}_y) \tag{2a}$$
$$\overrightarrow{\mathbf{h}}_t = \sigma(\overrightarrow{W}_x \mathbf{x}_t + \overrightarrow{W}_h \overrightarrow{\mathbf{h}}_{t-1} + \overrightarrow{\mathbf{b}}_h) \tag{2b}$$
$$\overleftarrow{\mathbf{h}}_t = \sigma(\overleftarrow{W}_x \mathbf{x}_t + \overleftarrow{W}_h \overleftarrow{\mathbf{h}}_{t+1} + \overleftarrow{\mathbf{b}}_h) \tag{2c}$$

BiRNNs as URNNs. Consider a Bidirectional RNN with the output nonlinearity ϕ defined as the softmax function $p_i = \phi_i(\mathbf{s}) = e^{s_i} / \sum_j e^{s_j}$ It is straightforward to show that the conditional probability of $\mathbf{y}_t$ given all other tokens can be written as

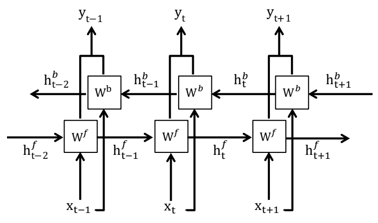
$$\begin{aligned}\mathrm{P}(\mathbf{y}_t \mid X_{[1:T]}) &= \phi(\overrightarrow{W}_y \overrightarrow{\mathbf{h}}_t + \overleftarrow{W}_y \overleftarrow{\mathbf{h}}_t + \mathbf{b}_y) \\ &\propto \phi\left(\overrightarrow{W}_y \overrightarrow{\mathbf{h}}_t + \frac{\mathbf{b}_y}{2}\right) \phi\left(\overleftarrow{W}_y \overleftarrow{\mathbf{h}}_t + \frac{\mathbf{b}_y}{2}\right)\end{aligned}$$

where the resulting terms in the proportionality resemble the URNNs output equation in Eq. 1a. Intuitively, this expression shows that the output of a Bidirectional RNN with a softmax output layer can be equivalently expressed as the product of the output from two independent but oppositely directed URNNs with specifically constructed weights, renormalized after multiplication. This construction also works in reverse such that an equivalent Bidirectional RNN can be constructed from two independently trained but oppositely directed URNNs. As such, we will consider a Bidirectional RNN as consisting of a forward-time model $\overrightarrow{\text{URNN}}$ and a backward-time model $\overleftarrow{\text{URNN}}$.

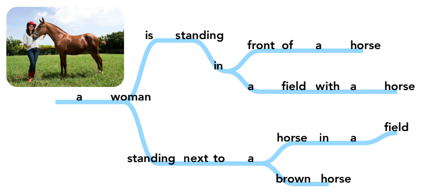
RNNs for decoding are trained to produce sequences conditioned on some encoded representation X. For machine translation tasks, X may represent an encoding of some source language sequence to be translated and Y is the translation. For image captioning, X is typically a dense vector embedding of the image produced by a Convolutional Neural Network (CNN) [25], and Y is a sequence of 1-hot encoding of the words of the corresponding image caption. Regardless of its source, this encoded representation is considered the first input $\mathbf{x}_0$ and for all remaining time steps $\mathbf{x}_t = \mathbf{y}_{t-1}$, such that decoder RNNs are learning to model $\mathrm{P}(\mathbf{y}_t | \mathbf{y}_{t-1}, ..., \mathbf{y}_1, \mathbf{x}_0)$. This is the setting of interest in this paper, but we drop this explicit dependence on the encoding $\mathbf{x}_0$ to reduce notational clutter in later sections.



(a) Unidirectional RNN



(b) Bidirectional RNN



(c) Example Run of Beam Search

Figure 2: **Different architectures of RNNs and left-to-right Beam Search.** (a) The prediction of variable $\mathbf{y}_t$ only depends on the *past* in URNNs. BiRNNs (b) can consider both *past* and *future*. (c) shows the search tree for beam search in a URNN with a beam width of B=4.

Beam Search (BS). Maximum a posteriori (MAP) (or more generally, M-Best-MAP [?,26]) inference in RNNs consists of finding the most likely sequence under the model. The primary difficulty for decoding is that the number of possible T length sequences grows exponentially as $|\mathcal{Y}|^T$, so approximate inference algorithms are employed. Due to this exponential output space and the dependence on previous outputs, exact inference is NP-hard in the general case. Beam Search (BS) is a greedy heuristic search algorithm that traverses the search tree using breadth-first search, while only expanding the most promising nodes at each depth. Specifically, BS in Unidirectional RNNs involves maintaining and expanding the top-B highest-scoring partial hypotheses, called *beams*. Let $Y_{[1:t]} = (\mathbf{y}_1, \dots, \mathbf{y}_t)$ denote a partial hypothesis (beam) at time t. We use the notation $\mathbf{Y}_{[1:B],[1:t]} = (Y_{1,[1:t]}, Y_{2,[1:t]}, \dots, Y_{B,[1:t]})$ to denote a collection of B beams. BS begins with empty beams, $Y_{b,0} = (\mathbf{y}_{b,0})$, where $\mathbf{y}_{b,0} = \emptyset$, $\forall b$ and proceeds in a left-to-right manner up to time T or until a special END token is generated. At each time t, BS considers the space of all possible beam extension $\mathcal{Y}_t = \mathbf{Y}_{[1:B],[1:t-1]} \times \mathcal{Y}$ and selects the top-B high-scoring t-length beams among this expanded hypothesis space. We can formalize this search for optimal updated beams $\mathbf{Y}_{[1:B],[1:t]}$ as

$$\underset{\mathcal{Y}_t}{\text{top-B}} \quad \log \mathrm{P}\left(Y_{[1:t]}\right) = \sum_{i=1}^{t} \log \mathrm{P}(\mathbf{y}_i \mid \mathbf{y}_{i-1}, \dots, \mathbf{y}_1).$$

Each log probability term in the above expression can be computed via a forward pass in Unidirectional RNNs such that implementing the top-B operation simply requires sorting $B|\mathcal{Y}_t|$ values. An example run of BS on a left-to-right URNN is shown in Figure 2c.

4. Bidirectional Beam Search (BiBS)

We begin by analyzing the decision made by left-to-right Beam Search at time t. Specifically, at each time t, we can factorize the joint probability $\mathrm{P}(Y_{[1:T]})$ in a particular way:

$$\mathrm{P}(Y_{[1,T]}) = \mathrm{P}(Y_{[1,t-1]})\mathrm{P}(\mathbf{y}_t|Y_{[1:t-1]})\mathrm{P}(Y_{[t+1:T]} \mid \mathbf{y}_t, Y_{[1:t-1]}) \tag{4}$$

This left-to-right decomposition of the joint around $\mathbf{y}_t$ is comprised of three terms

1) the 'marginal' of the sequence prior to $\mathbf{y}_t$: $\mathrm{P}(Y_{[1:t-1]})$,
2) the conditional of $\mathbf{y}_t$ given this past: $\mathrm{P}(\mathbf{y}_t|Y_{[1:t-1]})$, and
3) the conditional of the remaining sequence after $\mathbf{y}_t$ given all prior terms: $\mathrm{P}(Y_{[t+1:T]} \mid \mathbf{y}_t, Y_{[1:t-1]})$.

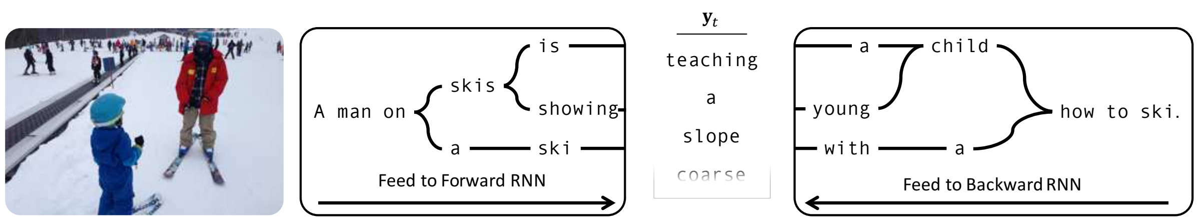
If we consider choosing $\mathbf{y}_t$ to maximize this joint, the first two terms can be computed exactly via the forward pass of a left-to-right URNN given the existing sequence; however, the third term cannot be exactly computed because it depends on all futures. Even approximating the third term with beams requires re-running beam search for each possible setting of $\mathbf{y}_t$, which is prohibitively expensive.

One way of interpreting left-to-right BS is to view it as approximating the joint in (4) with just the first two terms. Specifically, if we assume that $\mathrm{P}(Y_{[t+1:T]} \mid \mathbf{y}_t, Y_{[1:t-1]})$ is uniform, *i.e.* all futures are equally likely given the sequence so far, then BS is picking the optimal $\mathbf{y}_t$. This approximation does not hold in practice and results in poor performance for fill-in-the-blank tasks where all future sequences are not equally likely by design. In this section, we consider an alternative approximation and derive our BiBS approach.

Efficiently Approximating the Future. In order to derive a tractable approximation to this third term (and by proxy the full joint), we make two simplifying assumptions (which we know will be violated in practice, but result in an efficient approximate inference algorithm). First, we assume that future sequence tokens are independent of past sequence tokens given $\mathbf{y}_t$, *i.e.* RNNs are first-order Markov. Second, we assume that $\mathrm{P}(\mathbf{y}_t)$ is uniform, avoiding the need to estimate marginal distributions over $\mathcal{Y}$ for all time steps. Under these assumptions, we write the conditional probability of the remaining sequence tokens given the past sequence as

$$\begin{aligned} \mathrm{P}(Y_{[t+1:T]} \mid Y_{[1:t]}) &= \mathrm{P}(Y_{[t+1:T]} \mid \mathbf{y}_t) \\ &\propto \mathrm{P}(\mathbf{y}_t \mid Y_{[t+1:T]})\mathrm{P}(Y_{[t+1:T]}) \end{aligned} \tag{5}$$

Notice that the resulting terms are exactly the output of a right-to-left Unidirectional RNN. Substituting Eq. 5 into Eq. 4, we arrive at an expression that is proportional to the



$$Y_{[1:B],[1:t]} = \underset{b,\mathbf{y}_t,b'}{\text{top-B}} \quad \log \mathrm{P}(Y_{b,[1:t-1]}) \;+\; \log \left(\mathrm{P}\left(\mathbf{y}_t|Y_{b,[1:t-1]}\right)\mathrm{P}\left(\mathbf{y}_t \mid Y_{b',[t+1:T]}\right)\right) \;+\; \log \mathrm{P}(Y_{b',[t+1:T]}) \tag{3}$$

Figure 3: **Overview of Bidirectional Beam Search (BiBS).** Starting from a set of B complete sequences $Y_{[1,B],[1,T]}$, BiBS alternately performs left-to-right and right-to-left beam searches to greedily optimize an approximation of the probability of the entire sequence. In the example above, a left-to-right beam search is advancing the beams at time t by considering all possible connections between the current left-to-right beams and the previous right-to-left beams through any token in the dictionary $\mathcal{Y}$. The terms in this joint approximation (written in (3)) can be efficiently computed by the forward and backward Unidirectional RNNS and sorted to find the top-B extensions.

full joint, but comprised of terms which can be independently computed from a pair of oppositely-directed Unidirectional RNNs (or equivalently a Bidirectional RNN),

$$\underbrace{\mathrm{P}(Y_{[1:t-1]})\mathrm{P}(\mathbf{y}_t|Y_{[1:t-1]})}_{\text{Compute from }\overrightarrow{\text{URNN}}}\overbrace{\mathrm{P}(\mathbf{y}_t \mid Y_{[t+1:T]})\mathrm{P}(Y_{[t+1:T]})}^{\text{Compute from }\overleftarrow{\text{URNN}}} \tag{6}$$

Note that the two central conditional terms are proportional to the output of an equivalent softmax Bidirectional RNN as discussed in the previous section.

Coordinate Descent. Given some initial sequence $Y_{[1:t]}$, a simple coordinate descent algorithm could select a random time t and update $\mathbf{y}_t$ such that this approximate joint is maximized and repeat this until convergence. Computing Eq. 6 would require feeding $Y_{[1:t-1]}$ to the forward RNN and $Y_{[t+1:T]}$ to the backward RNN. Therefore, updating all outputs M times in this approach would require MT^2 RNN steps (combined from both the forward and backward models). If we instead follow an alternating left-to-right then right-to-left update order, this can be reduced to MT by reusing cached log probabilities from the previous direction. This algorithm resembles a beam search with $B = 1$ which bases extensions on the value of Eq. 6.

Bidirectional Beam Search. Finally, we arrive at our full Bidirectional Beam Search (BiBS) algorithm by generalizing the simple algorithm outlined above to maintain multiple beams during each update pass. Given some set of initial sequences $Y_{[1:B],[1:T]}$ (perhaps from a left-to-right beams search), we alternate between forward (left-to-right) and backward (right-to-left) beam searches with respect to the approximate joint. We consider a pair of forward and backward updates a single round of BiBS.

Without loss of generality, we will describe a forward update pass of beam width B. At each time t, we have updated the first $t-1$ tokens of each beam such that we have partial forward sequences $Y_{[1:B],[1:t-1]}$ and the values $Y_{[1:B],[t+1:T]}$ have yet to be updated. To update the forward beams, we consider all possible connections between the current left-to-right beams and the right-to-left beams (held fixed from previous round) through any token in the dictionary $\mathcal{Y}$. Our search space is then $\mathcal{Y}_t = Y_{[1:B],[1:t-1]} \times \mathcal{Y} \times Y_{[1:B],[t+1:T]}$ and $|\mathcal{Y}_t| = B \times |\mathcal{Y}| \times B$.

Figure 3 shows an example left-to-right update step for image captioning as well as the precise update rule based on Eqn. 6 for this time step. For each combination of forward beam and backward beam, this objective can be computed easily from stored sum of log probabilities of each beam and conditional output of the forward and backward RNNs. Like standard Beam Search, the optimal extensions can be found exactly by sorting these values for all possible combinations. Our approach requires only $2BMT$ RNN steps to perform M rounds of updates. Our algorithm is summarized below in Alg. 1 with $\overrightarrow{\theta}_{b,i}(\mathbf{y}_{b,i})$ representing $\log \mathrm{P}(\mathbf{y}_{b,i}|Y_{b,[1:i-1]})$ from the $\overrightarrow{URNN}$ and $\overleftarrow{\theta}_{b,i}(\mathbf{y}_{b,i})$ representing $\log \mathrm{P}(\mathbf{y}_{b,i}|Y_{b,[i+1:T]})$ from the $\overleftarrow{URNN}$.

```
Data: Given initial set of sequences $Y_{[1:B],[1:T]}$
$\overrightarrow{\theta}_{[1:B],[1:T]} = \overleftarrow{\theta}_{[1:B],[1:T]} = \mathbf{0}$
while not converged do
    // Updated beams left-to-right
    for $t = 1, ..., T$ do
        $\overrightarrow{\theta}_{[1:B],t},\ \overrightarrow{h}_{[1:B],t} = \overrightarrow{\text{URNN}}\left(\overrightarrow{h}_{[1:B],t-1}, Y_{[1:B],t-1}\right)$
        $Y_{[1:B],t} = \text{top-B} \sum_{i=1}^{t} \overrightarrow{\theta}_{b,i}(\mathbf{y}_{b,i}) + \sum_{j=t}^{T} \overleftarrow{\theta}_{b',j}(\mathbf{y}_{b',j})$
    end
    // Updated beams right-to-left
    for $t = T, ..., 1$ do
        $\overleftarrow{\theta}_{[1:B],t},\ \overleftarrow{h}_{[1:B],t} = \overleftarrow{\text{URNN}}(\overleftarrow{h}_{[1:B],t+1}, Y_{[1:B],t+1})$
        $Y_{[1:B],t} = \text{top-B} \sum_{i=t}^{T} \overleftarrow{\theta}_{b,i}(\mathbf{y}_{b,i}) + \sum_{j=1}^{t} \overrightarrow{\theta}_{b',j}(\mathbf{y}_{b',j})$
    end
end
```

Algorithm 1: Bidirectional Beam Search (BiBS).

5. Experiments

In this section, we evaluate the effectiveness of our proposed Bidirectional Beam Search (BiBS) algorithm for inference in BiRNNs. To examine the performance of bidirectional inference, we evaluate on tasks that require the generated sequence to fit well with existing structures both in the past and the future. We choose fill-in-the-blank style tasks where a number of tokens have been removed from a sequence and must be reconstructed. Specifically, we evaluate on fill-in-the-blank tasks on image captioning for the Common Objects in Context (COCO) [27] dataset and descriptions from the Visual Madlibs [28] dataset.

Baselines. We compare our approach, which we denote **`BiRNN-BiBS`**, against several sophisticated baselines:

- **`URNN-f:`** that runs BS on a forward LSTM to produce B output beams (ranked by their probabilities under the forward LSTM),
- **`URNN-b:`** that runs BS on a backward LSTM to produce B output beams (ranked by their probabilities under the backward LSTM),
- **`URNN-f+b:`** that runs BS on forward and backward LSTMs to produce $2B$ output beams (ranked by the maximum of the probabilities assigned by the forward and backward LSTMs). The method used by Wang *et al.* [21].
- **`BiRNN-f+b:`** that runs BS on two LSTMs (forward and backward) to produce $2B$ output beams (ranked by the sum of the log probabilities assigned by the forward and backward LSTMs). This lacks formal justification but we find it to be a reasonable heuristic for this task.
- **`GSN (Ordered):`** that samples tokens from the BiRNN for each time step. We found randomly selecting the time step as in [22] resulted in poor performance on our tasks and instead perform updates in an alternating left-to-right / right-to-left order. For fairness, we compare at the same number of updates as our method and all sample sequences are reranked based on log probability.

All baselines perform inference on the same trained model that we train using *neuraltalk2* [8] with standard maximum-likelihood training over complete human captions.

Evaluation. For all models, we evaluate only the top beam from the sorted list returned by the algorithm. We compare methods on standard sentence-level metrics – CIDEr [29], Meteor [30], and Bleu [31] – computed between the ground truth captions and the (full) reconstructed sentences. We note that the metrics are computed over the entire sentence (and not just the blank region) in order to capture the quality of the alignment of the generated text with the existing sentence structure. As a side effect, the absolute magnitude of these metrics are inflated due to the correctness of the context words, so we focus on the relative performance.

5.1. Fill-in-the-Blank Image Captioning on COCO

The COCO [27] dataset contains over 120,000 images, each with a rich set of annotations. This include five captions describing the content of each image, collected from Amazon Mechanical Turk workers. We propose a novel fill-in-the-blank image captioning based on this data. Given an image I and a corresponding ground truth caption $\mathbf{y}_1, ..., \mathbf{y}_T$ from the dataset, we remove a sequential portion of the caption such that we are left with a prefix $\mathbf{y}_1, \ldots, \mathbf{y}_s$ and suffix $\mathbf{y}_e, \ldots, \mathbf{y}_T$ consisting of the remaining words on either side of the blank. Using the image and the context from these remaining words, the goal is to generate the missing tokens $\mathbf{y}_{s+1}, \ldots, \mathbf{y}_{e-1}$. This is a challenging task that explores how well models and inference algorithms reason about the past and future during sequence generation. We first consider the known blank length setting (where the inference algorithm knows the blank length) and then generalize to the unknown blank length setting.

Known Blank Length. In this experiment, we remove $r =$ 25%, 50%, or 75% of the words from the middle of a caption for each image and task the model with generating the lost content. For example, at $r = 50\%$ the caption *"A close up of flowers and plants inside of a bowl"* would appear to the system as the blanked caption *"A close* __ __ ______ ___ ______ ______ *of a bowl"* and the generation task would then be to reproduce the removed subsequence of words *"up of flowers and plants inside."*

As we are interested in bidirectional *inference* (not learning), we train our models on the original COCO image captioning task (*i.e.* we do not explicitly train to fill blanked captions). Like [8], we use 5000 images for test, 5000 images for validation, and the rest for training. We evaluate on a single caption per image in the test set.

The upper half of Table 1 reports the performance of our approach (BiBS) on this fill-in-the-blank inference task for differently sized blanks (r% of central words removed per sentence). We run GSN and BiBS for four full forward / backward passes of updates. Generally we find that bidirectional methods outperform unidirectional ones on this task. We find that BiBS outperforms all baselines on all metrics. We note that the nearest baselines in performance (`URNN-f+b`, `BiRNN-f+b`) are reranked from 2B beams. While BiBS operates in an alternating left-to-right and right-to-left fashion, it only ever maintains B beams.

Interestingly, backward time model URNN-b consistently outperforms the forward time model URNN-f on all metrics and across all sizes of blanks. This may be due to the way the dataset was collected. When tasked with describing the content of an image, people often begin by grounding their sentences with respect to specific entities visible in the image (especially when humans are depicted). Given this, we would expect many more sentences to begin with the simi-

a) The woman has many bananas and other fruit at her stand
b) The woman has a bunch of bananas on at her stand
c) The woman has holding a bunch of bananas at her stand
d) The woman has a large bunch of bananas at her stand

a) A man is skateboarding on a ramp in a basement
b) A man riding a skateboard up the in a basement
c) A man a trick on a skateboard in a basement
d) A man doing tricks on a skateboard in a basement

a) A number of small planes behind a fence
b) A number of small planes on a fence
c) A number plane is parked near a fence
d) A number of planes parked near a fence

a) A black and yellow bird with a colorful beak
b) A black and yellow bird sitting a colorful beak
c) A black a yellow bird with a colorful beak
d) A black and yellow bird with in a basement

a) A group of people standing on top of a snow covered slope
b) A group of people on skis on a snowy snow covered slope
c) A group of riding skis on top of a snow covered slope
d) A group of people standing on top of a snow covered slope

a) A row of transit buses sitting in a parking lot
b) A row of buses parked in a a parking lot
c) A row of double decker buses parked a parking lot
d) A row of red buses parked in a parking lot

a) The person is riding the waves in the water
b) The person is is riding a wave in the water
c) The person is person on a surfboard in the water
d) The person is is on a surfboard in in the water

a) Two people riding a motorcycle to the beach
b) Two people on a motorcycle on the beach
c) Two people riding a motorcycle on the beach
d) Two people on a motorcycle on the beach

a) Ground Truth **b) URNN-f** **c) URNN-b** **d) BiRNN-BiBS**

Figure 4: **Example fill-in-the-blank image caption completions generated by BS and BiBS.** URNNs decoded with BS often produce blank reconstructions that clash with the remaining context on either side of the blank, while BiBS handles these transitions seamlessly.

		r=0.25			r=0.5			r=0.75		
		CIDEr	Bleu-4	Meteor	CIDEr	Bleu-4	Meteor	CIDEr	Bleu-4	Meteor
Known Length	URNN-f	6.54	0.661	0.488	3.744	0.345	0.350	1.927	0.143	0.238
	URNN-b	6.58	0.668	0.491	3.931	0.372	0.356	2.476	0.219	0.259
	URNN-f+b [21]	6.98	0.709	0.510	4.15	0.398	0.367	2.40	0.209	0.257
	BiRNN-f+b	6.94	0.705	0.508	3.99	0.385	0.361	2.24	0.201	0.252
	GSN [22] (Ordered)	6.90	0.701	0.507	3.63	0.337	0.334	1.876	0.135	0.232
	BiRNN-BiBS (ours)	**7.12**	**0.720**	**0.517**	**4.26**	**0.408**	**0.368**	**2.57**	**0.228**	**0.265**
Unknown Length	URNN-f	5.607	0.569	0.440	4.232	0.432	0.370	2.594	0.268	0.269
	URNN-b	5.514	0.561	0.436	4.151	0.424	0.367	2.909	0.303	0.285
	URNN-f+b [21]	5.632	0.570	0.440	4.377	0.451	0.376	2.924	**0.306**	0.287
	BiRNN-f+b	5.640	0.588	0.452	4.380	0.453	0.378	2.930	0.305	0.303
	GSN [22] (Ordered)	5.725	0.589	0.447	3.591	0.413	0.357	2.456	0.257	0.261
	BiRNN-BiBS (ours)	**5.935**	**0.614**	**0.460**	**4.40**	**0.454**	**0.380**	**2.936**	0.305	**0.288**

Table 1: **Comparison of different approaches on Fill-in-the-Blank Image Captioning on COCO [27].** r is the fraction of removed words from sentence, B=5 by default. BiBS consistently outperforms the baselines methods.

lar words such that generating the beginning of a sentence from the end would be an easier task.

BiBS Convergence. To study the convergence of our approach, we consider the true joint probability of filled-in captions as a function of the number of rounds of BiBS. We compute the average of these joint log probabilities after each meta-iteration of BiBS, where we define a meta-iteration as a pair of full forward and backward update passes. We find that joint log probabilities drop quickly (reducing from -2.47 to -2.11 in a single meta-iteration), indicating high quality solutions are found from unidirectional initializations within only a few meta-iterations of BiBS. In practice we find the beams have converged in typically 1 to 2 meta-iterations for fill-in-the-blank image captioning.

Fig. 4 shows several qualitative examples, comparing completed captions from `URNN-f`, `URNN-b`, and our `BiRNN-BiBS` method with ground truth human annotations. The unidirectional models running standard BS typically generate sentences that abruptly clash with existing words at the edge of the blank. For example in the first example, the forward model produces the grammatically incorrect phrase "bananas on at her stand" and similarly the backward model outputs "The woman has holding a bunch". This behavior is a natural consequence of the inability for these models to efficiently reason about the past and future simultaneously. While these unidirectional models struggle to reason about word transitions on either side of the blank, our BiRNN based BiBS algorithm typically produces reconstructions that smoothly fit with the context, producing a reasonable sentence "The woman has a large bunch of bananas at her stand."

This example also highlights a possible deficiency in our evaluation metrics; while a human observe can clearly tell which of the three sentences is most natural, the sentence level statistics of each are quite similar, with each sharing only the word banana with the ground truth caption "The woman has many bananas and other fruit at her stand." Evaluating generated language is a difficult and open problem that is further complicated by fill-in-the-blank context.

Unknown Length Blanks. While our method is designed for known blank lengths, in this section we apply BiBS as a black box inference algorithm over a range of blank lengths. We calibrate what lengths to search over by first generating the top-1 left-to-right beam Y^f by only conditioning on words on the left of the blank and the right-to-left top-1 beam Y^b by only conditioning words on the right side of the blank. Then, we define the range of lengths of the blank as $\min\{len(Y^f), len(Y^b)\}$ to $\max\{len(Y^f), len(Y^b)\}$ where, $len(Y)$ is the length of beam Y. We perform inference at each length in this range and select the highest probability completion across all lengths. The lower half of Table 1 reports the results. We find that BiBS outperforms nearly all baselines on all metrics (narrowly being bested by `URNN-f+b` at $r = 0.75$ Blue-4). Results on the variable length task are worse for all methods than with known blank length due largely to the difficulty of comparing the likelihoods of sequences with differing lengths.

5.2. Visual Madlibs

In this section, we evaluate our approach on the Visual Madlibs [28] fill-in-the-blank description generation task. The Visual Madlibs dataset contains 10,738 images with 12 types of fill-in-the-blank questions answered by 3 workers on Amazon Mechanical Turk. We use *object's affordance* (type 7) and *pair's relationship* (type 12) fill-in-the-blank questions as these types have blanks in the middle of questions. For instance, *People could relax on the couches.* and *The person is putting food in the bowl.* We use 2000 images for validation, train on the remaining images from the train set, and evaluate on their 2,160 image test set. To the best of our knowledge, ours is the first paper to explore the performance of CNN+LSTM text generation for this task.

We compare to two additional baselines for these experiments, nCCA [32] and the nCCA(box) method implemented in the Visual Madlibs paper [28]. nCCA maps image and text to a joint-embedding space and then finds the nearest neighbor from the training set to this embedded

	type 7		type 12	
	Bleu-1	Bleu-2	Bleu-1	Bleu-2
URNN-f	0.313	0.138	0.275	0.160
URNN-b	0.460	0.284	0.346	0.213
URNN-f+b [21]	0.447	0.275	0.347	0.214
BiRNN-f+b	0.448	0.275	0.347	0.213
GSN [22] (Ordered)	0.427	0.28	0.148	0.099
BiRNN-BiBS (ours)	**0.470**	**0.389**	**0.353**	**0.216**
nCCA	0.56	0.1	0.46	0.07
nCCA(box)	0.60	0.11	0.48	0.08

Table 2: **Comparison of different approaches on the Visual Madlibs task** using BLEU-1 and BLEU-2. B= 5 by default.

point. We note that this is a retrieval and not a description generation technique such that it cannot be directly compared with BiBS, and is reported only for the sake of completeness. nCCA(box) extracts visual features from the ground-truth bounding box of the relevant person or object refered to in the question and thus is an 'oracle' result that makes use of extra ground truth information.

We again use the *neuraltalk2* [8] framework to train a CNN+LSTM model for both *object's affordance* (type 7) and *pair's relationship* (type 12) question types. We evaluate on the test data using Bleu-1 and Bleu-2 (to be consistent with [28]). Table 2 shows the results of this experiment for known blank length (see supplementary for unknown length results). We find that BiBS outperforms the other generation based baselines in both question types and is competitive with the retrieval based nCCa techniques, greatly outperforming the nCCa retrieval and nCCA(box) oracle methods on Bleu-2.

6. Conclusions

In summary, we presented the first approximate inference algorithm for 1-Best (and M-Best) decoding in bidirectional neural sequence models (RNNs, LSTMs, GRUs, *etc*.). We study our method in the context of a novel fill-in-the-blank image captioning task which evaluates how well sequence generation models and their associated inference algorithms incorporate known information from both the past and future to 'fill in the gaps'. This is a challenging setting and we demonstrate that standard Beam Search is poorly suited for this task. We develop a Bidirectional Beam Search (BiBS) algorithm based on an approximation of the full joint distribution over output sequences that is efficient to compute in Bidirectional Recurrent Neural Network models. To the best of our knowledge, this is the first algorithm for top-B MAP inference in Bidirectional RNNs. We have demonstrated that BiBS outperforms natural baselines at both fill-in-the-blank image captioning and Visual Madlibs. Future work involves generalizing these ideas to tree-structured or more general recursive neural networks [33], and producing diverse M-Best sequences [34, 35].

Acknowledgements We thank Rama Vedantam for initial brainstorming. This work was funded in part by the following awards to DB: NSF CAREER, ONR YIP, ONR Grant N00014-14-1-0679, ARO YIP, and NVIDIA GPU donations. SL was supported in part by the Bradley Postdoctoral Fellowship. Conclusions contained herein are the authors and are not to be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[5] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models.," in *EMNLP*, 2013.

[6] O. Vinyals and Q. V. Le, "A neural conversational model," *http://arxiv.org/pdf/1506.05869v3.pdf*, 2015.

[7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *CVPR*, 2015.

[8] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015.

[9] H. Fang, S. Gupta, F. N. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig, "From Captions to Visual Concepts and Back," in *CVPR*, 2015.

[10] X. Chen and C. L. Zitnick, "Mind's Eye: A Recurrent Visual Representation for Image Caption Generation," in *CVPR*, 2015.

[11] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," in *CVPR*, 2015.

[12] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *ICCV*, 2015.

[13] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *NIPS*, 2015.

[14] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning," *CVPR*, 2017.

[15] M. Malinowski and M. Fritz, "A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input," in *NIPS*, 2014.

[16] D. Geman, S. Geman, N. Hallonquist, and L. Younes, "A Visual Turing Test for Computer Vision Systems," in *PNAS*, 2014.

[17] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. Moura, D. Parikh, and D. Batra, "Visual dialog," *CVPR*, 2016.

[18] H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. C. Courville, "Guesswhat?! visual object discovery through multi-modal dialogue," *CVPR*, 2017.

[19] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *CoRR*, 2015.

[20] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *arXiv preprint arXiv:1507.06947*, 2015.

[21] C. Wang, H. Yang, C. Bartz, and C. Meinel, "Image captioning with deep bidirectional lstms," *CoRR*, 2016.

[22] M. Berglund, T. Raiko, M. Honkala, L. Karkkainen, A. Vetek, and J. Karhunen, "Bidirectional recurrent neural networks as generative models," in *NIPS*, 2015.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, 1997.

[24] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] D. Batra, "An Efficient Message-Passing Algorithm for the M-Best MAP Problem," in *UAI*, 2012.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," 2014.

[28] L. Yu, E. Park, A. C. Berg, and T. L. Berg, "Visual Madlibs: Fill in the blank Description Generation and Question Answering," *ICCV*, 2015.

[29] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *CVPR*, pp. 4566–4575, 2015.

[30] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.

[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 311–318, 2002.

[32] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 210–233, 2014.

[33] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *ICML*, 2011.

[34] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich, "Diverse M-Best Solutions in Markov Random Fields," in *ECCV*, 2012.

[35] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. J. Crandall, and D. Batra, "Diverse beam search: Decoding diverse solutions from neural sequence models," *arXiv*, vol. abs/1610.02424, 2016.