Trainable Decoding of Sets of Sequences for Neural Sequence Models

Ashwin Kalyan [1] Peter Anderson [1] Stefan Lee [1] Dhruv Batra [1 2]

Abstract

Many sequence prediction tasks admit multiple correct outputs and so, it is often useful to decode a set of outputs that maximize some task-specific set-level metric. However, retooling standard sequence prediction procedures tailored towards predicting single best outputs tends to produce sets containing very similar sequences; failing to capture the variation in the output space. To address this, we propose ∇BS, a trainable decoding procedure that outputs a set of sequences, highly valued according to the metric. Our method tightly integrates the training and decoding phases and further allows for the optimization of the task-specific metric addressing the shortcomings of standard sequence prediction. Further, we discuss the trade-offs of commonly used set-level metrics and motivate a new set-level metric that naturally evaluates the notion of "capturing the variation in the output space". Finally, we show results on the image captioning task and find that our model outperforms standard techniques and natural ablations.

1. Introduction

Given an input $\mathbf{x}$, sequence prediction problems require outputting a single sequence $\mathbf{y}$ that it is highly valued as measured by some task specific metric $\phi(\mathbf{y}|\mathbf{x})$; for example, BLEU (Papineni et al., 2002) is a commonly used metric for language generation tasks. However, many real-world sequence prediction problems are inherently multimodal *i.e*. for a given input, there can be multiple outputs $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_K\}$ that are highly valued according to the metric. For instance, the task of image captioning (Chen et al., 2015) admits multiple correct outputs because an image can be accurately described in numerous ways by focusing on different objects and interactions present in the image (Kalyan et al., 2018). Being able to produce multiple relevant output sequences is not only important from a modeling perspective, but is also beneficial even in tasks in which a single best output is ultimately required. For example, in the task of automated response suggestion for email, Kannan et al. (2016) allow the user to select from a set of generated responses with different sentiments. Similarly, producing multiple outputs and then reranking them leads to improvements in the task of machine translation (Shen et al., 2004; Li et al., 2016). In these settings where a set of sequences is expected, the quality of the generated set is measured using set-level metrics $\Phi(\mathcal{Y}|\mathbf{x})$ that evaluate higher-order interactions between elements of the decoded set. For example, *oracle* accuracy is a set-level metric that corresponds to the maximum sequence level score achieved by any of the sequences in the generated set. It is commonly used as a proxy for a downstream selection mechanism (Zhang et al., 2006; Batra et al., 2012).

Decoding K outputs using Sequence Models. In practice, the standard single-sequence prediction pipeline can be used to produce a set of K outputs. In this setup, neural sequence models like RNNs, LSTMs (Hochreiter & Schmidhuber, 1997) or Transformers (Vaswani et al., 2017) trained to maximize the likelihood of *individual* sequences are used in conjunction with approximate top-K inference procedures like Beam Search (BS). As the goal of this procedure is to find the single best output, BS does not consider intra-set interactions in the decoded output set. Naturally, this leads to the decoding of largely redundant output sets containing near identical sequences (Vijayakumar et al., 2018; Li et al., 2016; Jiang & de Rijke, 2018). While the specific issue of diversity has been addressed by a variety of approaches that either modify the training objective (Dai & Lin, 2017; Wang et al., 2017), learn model ensembles (Lee et al., 2016; Wang et al., 2016; He et al., 2018) or modify the inference procedure (Vijayakumar et al., 2018; Li et al., 2016), these methods are incapable of modeling higher-order interactions between the sequences in the decoded set and by extension, cannot optimize arbitrary set-level metrics.

Trainable Decoding of Sequence Sets. In this work,

[1] School of Interactive Computing, Georgia Tech, Atlanta, GA, USA [2] Facebook AI Research, Menlo Park, CA, USA. Correspondence to: Ashwin Kalyan <ashwinkv@gatech.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

we propose ∇BS[1], a trainable decoder that finds approximate solutions for the best *set* of K sequences by accounting for intra-set interactions. Our approach directly models a set of outputs and allows for maximizing both the set-level metric of interest, or the likelihood of a target set when multiple ground truth annotations are provided. We achieve this by treating the task as a sequential subset selection problem, a novel perspective that allows us to utilize techniques from the well-studied problem of cardinality-constrained submodular maximization (Nemhauser et al., 1978). Our method closely mimics BS; replacing the likelihood informed pruning of the search space with a subset selection step that is guided by a *learned* submodular set function. Unlike existing sequence models, our approach considers intra-set interactions and induces a distribution over sets of sequences allowing the use of greedy decoding to find the the maximizer set of size K.

Contributions. In summary, the primary technical contribution of our work is ∇BS, a *task-agnostic trainable* decoding procedure for *sets of sequences*. In the context of the proposed decoder, we discuss various training strategies inspired from both supervised learning and reinforcement learning that ensure stable training while mitigating loss-evaluation mismatch and exposure bias (Ranzato et al., 2015). Further, we motivate a new set-level metric inspired by the facility location problem (Stollsteimer, 1963) that naturally evaluates the notion of "capturing the variation in the output space". Finally, we choose the popular sequence prediction task of image captioning to demonstrate the effectiveness of our method and find that our approach, ∇BS consistently outperforms standard techniques and ablations of our method on relevant set-level metrics.

2. Approach

We are interested in predicting a set of K sequences $\mathcal{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \dots \mathbf{y}^K\}$ given some input $\mathbf{x}$ such that $\mathcal{Y}$ is highly valued according to some set-level metric $\Phi(\mathcal{Y}|\mathbf{x})$. While neural sequence models have been used to address this problem in conjunction with decoding strategies like Beam Search, existing approaches can neither learn intra-set interactions nor optimize for arbitrary set-level metrics. In this work, we propose ∇BS, a novel trainable set decoding procedure capable of modeling interactions between elements of the set. Further, our approach tightly integrates the training and decoding phases overcoming exposure bias or loss-evaluation mismatch suffered by standard sequence models.

In the remainder of this section, we discuss how decoding a set of sequences can be viewed as sequential subset selection problem. We then show how this problem can be parameterized with a submodular function and then trained via gradient descent.

2.1. Preliminaries: Sequence Prediction and Beam Search Decoding

For convenience, let $[n]$ denote $\{1, 2, \dots n\}$ and $\mathbf{v}_{\leq n}$ denote $\{v_1, v_2, \dots v_n\}$, the first n elements of a vector $\mathbf{v} \in \mathbb{R}^m$.

Sequence prediction problems require outputting a sequence $\mathbf{y}$ given an input $\mathbf{x}$ such that $\mathbf{y}$ is highly valued according to some task specific metric $\phi(\mathbf{y}|\mathbf{x})$. The common approach to these sorts of problems is to learn a transition function $\mathbb{P}_\theta(y_t|\mathbf{y}_{\leq t-1}, \mathbf{x})$ parametrized by $\theta \in \mathbb{R}^d$ that represents the probability of choosing the next token from a vocabulary $\mathcal{V}$ given the previous tokens $\mathbf{y}_{\leq t-1}$ and the initial input $\mathbf{x}$. Given such a model, the most likely output sequence under the model can be found by solving:

$$\mathbf{y}^* = \operatorname{argmax} \mathbb{P}(\mathbf{y}|\mathbf{x}) \tag{1}$$

$$= \operatorname{argmax} \prod_{t \in [T]} \mathbb{P}_\theta(y_t|\mathbf{y}_{\leq t-1}, \mathbf{x}) \tag{2}$$

Due to the exponential size of the output space ($|\mathcal{V}|^T$ possibilities for a T length sequence), this inference problem is NP-hard. Therefore, greedy heuristics are used to find approximate solutions in practice – for example, argmax decoding greedily selects the most likely token at each time step to find the approximate single best solution.

Beam Search (BS), a widely-used approximate inference algorithm provides an alternative to argmax decoding. BS maintains a set of K partial solutions, and often yields better approximate solutions than argmax decoding as it explores a slightly larger portion of the search space (argmax decoding is BS with $K = 1$). BS operates left-to-right in a greedy manner – at each time step $t \in [T]$, using the current set of K partial solutions, $\mathcal{Y}_{t-1} = \{\mathbf{y}^k_{\leq t-1}\}_{k \in [K]}$, BS constructs the next set $\mathcal{Y}_t$ as:

$$\{\mathbf{y}^k_{\leq t}\}_{k \in [K]} = \underset{\mathcal{Y}_{t-1} \times \mathcal{V}}{\text{argmax-K}} \sum_{k \in [K]} \prod_{t \in [T]} \mathbb{P}(y^k_t|\mathbf{y}^k_{\leq t-1}, \mathbf{x}) \tag{3}$$

where argmax-K selects the K maximizing alternatives. This problem is trivially solved by ranking all the possible extensions, $\mathcal{Y}_{t-1} \times \mathcal{V}$ by their likelihood and selecting the K most likely elements. This is repeated at each time step until termination, at which point the most likely sequence is selected from the K decoded sequences.

2.2. Decoding As Sequential Subset Selection

Taking a high-level view, each step of Beam Search (BS) decoding performs a subset selection that is informed by the

[1] pronounced diff-BS, code available at https://github.com/ashwinkalyan/diff-bs

likelihood of sequences under the trained model – selecting the K most likely sequences from the $K \times |\mathcal{V}|$ options. Unlike likelihood that guides BS to perform this subset selection, it is often a strong requirement for the metric $\Phi(\cdot|\mathbf{x})$ to decompose across time steps in a similar manner; ruling out a naïve greedy decoder like BS informed by the set-level metric. Therefore, we are instead learning to select subsets, *i.e.* solve the $\binom{K\times|\mathcal{V}|}{K}$ problem of selecting the K most promising alternatives such that the resulting set $\mathcal{Y}_T$ after T time steps is highly valued by the set-level task metric $\Phi(\cdot|\mathbf{x})$.

Submodular Functions and Sequence-level Metrics. Task-specific metrics typically evaluate a notion of coverage *i.e. highly valued* outputs must overlap significantly with the "correct" outputs. For example – at a high level, metrics for language generation tasks evaluate a candidate sentence by checking for shared n-grams with a reference sentence. Submodular functions, an important class of set functions, elegantly capture the notion of coverage and therefore have not only motivated the development of popular sequence-level metrics (Hong et al., 2014) but some previously proposed metrics have been showed to belong to this function family (Lin & Bilmes, 2011). With the notion of coverage guiding the development of sequence-level metrics, various simple set-level metrics like average or maximum of individual sequence level-scores can also be shown to be submodular (as submodularity is preserved by these operations). While it may not possible to always show that task-specific metrics are exactly submodular, it can be expected that they are at least approximately submodular. This link between submodular functions and set-level metrics motivates us to develop a subset selection mechanism that uses *submodular maximization* at its core.

Before explaining our method in its entirety, we provide a brief overview of submodular functions and explain the classic greedy algorithm for maximizing them in the presence of cardinality constraints.

Submodular Maximization. Given a ground set $\mathcal{V}$, a set function $f : 2^{\mathcal{V}} \to \mathbb{R}_{\geq 0}$ assigns a value for all sets $\mathcal{S} \subseteq \mathcal{V}$. Finding a subset of some bounded size K that maximizes the set function *i.e.* $\operatorname{argmax}_{\mathcal{S}\subseteq\mathcal{V},|\mathcal{S}|\leq K} f(\mathcal{S})$ is a natural way of characterizing various coverage problems – for example, finding where to place K sensors such that the covered area as measured by f is maximized. Despite its usefulness, this maximization is NP hard for arbitrary functions. However, the classic result of Nemhauser et al. (1978) shows that a greedy strategy achieves a constant factor approximation ratio of $(1 - 1/e)$ if the function f is *monotone submodular*. Given sets $\mathcal{S}, \mathcal{T}$ *s.t.* $\mathcal{S} \subseteq \mathcal{T} \subseteq \mathcal{V}$ and $e \in \mathcal{V}\backslash\mathcal{T}$, a set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is submodular if

$$f(\mathcal{S} \cup \{e\}) - f(\mathcal{S}) \geq f(\mathcal{T} \cup \{e\}) - f(\mathcal{T})$$

i.e. adding an element to a larger set results in smaller gains; capturing the notion of diminishing returns. The *marginal utility* of adding a new element to the set (*e.g.* increase in coverage by placing a sensor) is given by the difference, $f(S \cup \{e\}) - f(S)$ which we denote by $\Delta_f(e|S)$. Further, the function f is (a) *monotone* if $f(\mathcal{S}) \leq f(\mathcal{T}), \forall \mathcal{S} \subseteq \mathcal{T} \subseteq \mathcal{V}$ and (b) *normalized* if $f(\emptyset) = 0$. The greedy strategy of Nemhauser et al. (1978) adds the element with the largest marginal gain at each step *i.e.*

$$A^k \leftarrow A^{k-1} \cup \underset{e\in\mathcal{V}\backslash A^{k-1}}{\operatorname{argmax}} \Delta_f(e|A^{k-1})$$

to yield A^K *s.t.* $f(A^K) \geq (1 - 1/e)\, f(A^*)$ after K steps.

Learning Subset Selection. Provided a submodular function that estimates the utility of the set chosen w.r.t. to maximizing the final set-level metric, we can construct a set-level policy to find the approximate maximizer using the greedy algorithm. Since the sequence-level metric does not decompose over time steps, the choice of a submodular function that can estimate the utility of a partial solution is not obvious. Following Tschiatschek et al. (2018), we choose to learn an appropriate function maximizing which yields good approximate solution sets. Further, they show that this maximization can be made differentiable by replacing the argmax operation by a $\operatorname{softmax}_\tau$ operation with temperature $\tau > 0$ and iteratively sampling each element[2] proportional to $\exp(\Delta_f(e|A_{i-1})/\tau)$ – yielding an updated approximation ratio of $1 - 1/e - \epsilon(\tau)$, where $\epsilon(\tau)$ is some decreasing function of the temperature.

Algorithm 1 Sequential Subset Selection

```
input: $f_\beta, \mathbf{x}, \mathcal{V}, \tau, K, T$
output: $\mathcal{Y}_T = \{\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_K\}$
$\mathcal{Y}_0 \rightarrow \emptyset$
for $t \in [T]$ do
    $\mathcal{C}_t \leftarrow \mathcal{Y}_{t-1} \times \mathcal{V}$
    $\mathcal{S}^0 \leftarrow \emptyset$
    for $k \in [K]$ do
        $\mathbf{g}^k[i] \leftarrow \Delta_{f_\beta}(c|\mathcal{S}^{k-1}), \forall c \in \mathcal{C}_t\backslash\mathcal{S}^{k-1}$
        $s^k \sim \operatorname{softmax}\left(\mathbf{g}^k/\tau\right)$
        $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \{s^k\}$
    end
    $\mathcal{Y}_t = \mathcal{S}^K$
end
return $\mathcal{Y}_T$
```

Sequential Subset Selection. Finally, given a submodular function f_β parametrized by β, a suitable temperature τ and other inputs necessary to perform BS, a straightforward algorithm for sequential subset selection can be written down (Algorithm 1); with bounded

[2] instead of selecting the one with the highest marginal gain

approximation error for each time step. At each time step t, given the set of partial sequences $\mathcal{Y}_{t-1}$, all possible extensions $\mathcal{C}_t = \mathcal{Y}_{t-1} \times V$ are produced and sampled from sequentially. Specifically, the k^{th} sequence s_y^k is sampled according to $\exp(\Delta_f(\cdot|\mathcal{S}_t^{k-1})/\tau)$ and added to a working set $\mathcal{S}_t^{k-1}$ such that $\mathcal{S}_t^k = \mathcal{S}_t^{k-1} \cup \{s_t^k\}$. This sampling procedure additionally allows us to compute the likelihood of the alternatives chosen at time t, $\mathbb{P}_f(\mathcal{Y}_t|\mathcal{Y}_{t-1}, \mathbf{x})$ as:

$$\prod_{k \in K} \frac{\exp(\Delta_f(s_t^k|\mathcal{S}_t^{k-1}))}{\sum_{s \in \mathcal{C}_t \setminus \mathcal{S}_t^{k-1}} \exp(\Delta_f(s|\mathcal{S}_t^{k-1}))} \tag{4}$$

If the order of the elements in the set does not matter, the probabilities of all permutations must be summed; however, we simply multiply by $K!$ to approximate this quantity (Tschiatschek et al., 2018). Combined with this sampling procedure, any given f implies a probability distribution on all full-length sequence subsets $\mathcal{S} \in 2^{\mathcal{Y}}$ suitable for the policy π, namely

$$\mathbb{P}_f(\mathcal{Y}_T|\mathbf{x}) = \prod_{t \in T} \mathbb{P}_f(\mathcal{Y}_t|\mathcal{Y}_{t-1}, \mathbf{x}) \tag{5}$$

Connection to Sequence Level Training. In the restricted setting of argmax or greedy decoding (BS with $K = 1$), Ranzato et al. (2015), Gu et al. (2017), *etc.* learn a policy $\pi(\cdot|\mathbf{x})$ such that acting according to it maximizes the *sequence level metric* $\phi(\cdot|\mathbf{x})$ *i.e.*

$$\pi^* = \operatorname*{argmax}_{\pi \in \Pi} \mathbb{E}_{(y_1, \ldots y_T) \sim \pi(\cdot|\mathbf{x})} \left[\phi(\mathbf{y}|\mathbf{x})\right]$$

Following the probabilistic interpretation of the submodular maximization procedure shown in (5), our approach lifts this greedy decoding strategy to reason about sets and thus, handle "beam search" ($K > 1$) *i.e.*

$$\pi^* = \operatorname*{argmax}_{\pi \in \Pi} \mathbb{E}_{(\mathcal{Y}_1, \ldots \mathcal{Y}_T) \sim \pi(\cdot|\mathbf{x})} [\Phi(\mathcal{Y}|\mathbf{x})] \tag{6}$$

As we will see later in Section 2.4, this connection allows us to come up with different strategies to train our model in different scenarios.

With this formulation, learning to predict sets of sequences that maximize the set-level metric requires learning a suitable monotone submodular function f.

2.3. Learning a Submodular Selection Policy

We would like to learn a monotone, submodular function $f_\beta : 2^{\mathcal{Y}} \to \mathbb{R}$ parameterized by β such that sampling from the policy induced by its maximization as described above maximizes the set-level task metric $\phi(\mathcal{Y}|\mathbf{x})$, that is to say

$$f_\beta = \operatorname*{argmax}_{f} \mathbb{E}_{\mathcal{Y} \sim \pi_f} [\Phi(\mathcal{Y}|\mathbf{x})] \tag{7}$$

In this section, we discuss the form and training of f_β.

Parameterizing Submodular Function f. We apply recent work on deep submodular function (DSF) modeling from (Zaheer et al., 2017a; Bilmes & Bai, 2017) to construct f. To familiarize the reader with this work, we note the key result that given non-negative input features x_+, a monotone submodular function can be parameterized by a neural network of arbitrary depth provided it consists of multiplication operations with non-negative weights and element-wise non-decreasing concave activation functions. We encourage readers to see these works for full details.

Constructing parametrized submodular functions requires suitable representations of sets. At each time step t, the set of partial solutions $\mathcal{Y}_{t-1} = \{y_1^t, \ldots, y_K^t\}$ are represented by their hidden states from an LSTM, *i.e.* $h_t^k = LSTM(y_k^t)$ and each token in V is represented by its corresponding word-vector $\mathbf{v}_t \in \mathbb{R}^d$. Therefore, each alternative $c_t \in \mathcal{C}_t = \mathcal{Y}_{t-1} \times \mathcal{V}$ can now be represented by a concatenation of these two representations, $[\mathbf{h}_t, \mathbf{v}_t]$. Given a set $\mathcal{S} \subseteq \mathcal{C}_t$, we compute a permutation invariant set representation as

$$\psi_\beta(\mathcal{S}) = \sum_{c_t \in S} \{\text{MLP}\left([\mathbf{h}_t, \mathbf{v}_t]\right)\}_+ \tag{8}$$

using an MLP followed by a ReLU non-linearity (denoted by $\cdot_+$) to ensure non-negativity of the features. Importantly, the bias of the MLP is set to 0 to ensure that the submodular function is normalized by construction[3]. The submodular function f_β is now defined similar as a two-layer DSF with the element-wise non-negative monotone concave function $\sigma(\cdot) = \log(1 + \cdot)$,

$$f_\beta(S) = \mathbf{w}_2^\top \sigma(W_1^\top \sigma(\psi_\beta(S))) \tag{9}$$

where $W_1 \in \mathbb{R}_{\geq 0}^{d \times m}$ and $\mathbf{w}_2 \in \mathbb{R}_{\geq 0}^m$. The parameters of a DSF can be learnt via gradient descent using automatic differentiation, similar to deep networks. However, the weights need to be non-negative, so an additional projection step is required which we denote by $\Pi_{\geq 0}$. In practice, evaluating the submodular function for all the elements in the ground set $K \times \mathcal{V}$ can be slow (for *e.g.* $|\mathcal{V}|$ in the case of COCO captioning task is ∼10000). In such cases, a standard sequence model can be used to first coarsely select the top $K' > K$ elements.

Connection to DivMBest. Allowing the function f to be arbitrary and not restricting it to be submodular, modifies our approach to a learnable variant[4] of Diverse

[3] The initial hidden state representing no history and the dummy input (*e.g.* start token) are both represented by vectors of all zeros.

[4] Removing the inductive bias and using standard MLPs instead of DSFs leads to worse performance; more details in the supplement.

Beam Search (DBS) (Vijayakumar et al., 2018). Extending DivMBest (Batra et al., 2012) to sequence models, DBS greedily selects K alternatives at each step by adding diversity constraints after each element is selected. While hand-crafted diversity penalties (*e.g.* hamming or n-gram distance based diversity) are used in DBS, this penalty is instead learned by the set function f.

2.4. Training A DSF for Set Decoding.

In this subsection, we discuss various training strategies to obtain good subset selection policies in practice.

Cross Entropy Loss. For many sequence prediction tasks, datasets contain multiple correct outputs – for instance, image captioning datasets like COCO (Lin et al., 2014) have five captions per image. In this case, the policy can be trained via teacher-forcing *i.e.* the cross-entropy loss of the model's predictions and the "ground-truth" subset is minimized at each time step. For example, if the oracle chooses subset $\mathcal{Y}_t^* = \{\mathbf{y}_{<t}^k\}_{k\in K}$ at time $t \in [T]$, then the the policy incurs the loss:

$$L(\pi) = -\sum_{t\in[T]} \log \mathbb{P}(\mathcal{Y}_t^*|\mathcal{Y}_{t-1}^*, \mathbf{x})$$

This method, which we denote by CE, is applicable only when multiple annotations are available during training.

REINFORCE. Unlike CE, this strategy directly optimizes for the task-specific metric and only requires the ability to query the metric. This strategy, denoted by RE, minimizes the objective in (5) by computing the gradient using REINFORCE (Williams, 1992) as:

$$\nabla_\beta J(\pi) = \mathbb{E}_{\mathcal{Y}_1,\ldots\mathcal{Y}_T}\left[(\phi(\mathcal{Y}|\mathbf{x})-b)\sum_{t\in T}\nabla_\beta \log \mathbb{P}(\mathcal{Y}_t|\mathcal{Y}_{t-1})\right]$$

Here, b is a baseline reward that is subtracted to reduce the variance in the gradient estimates (Greensmith et al., 2004). For example, choosing the baseline to be the value achieved by beam search ensures that the learnt policy is competent w.r.t. to it (Rennie et al., 2017). While this training strategy fixes both exposure bias and loss-evaluation mismatch, it suffers from noisy gradients despite using suitable baselines leading to poor convergence properties.

Queriable Expert. In many scenarios where output sets need to be produced, only one ground truth annotation may be available; ruling out the use of CE. As training via RE is extremely unstable, Imitation Learning strategies like DAgger (Ross et al., 2011) that use a queriable expert are often employed to *warm start* the policy. This setting can be used to warm-start the set-level decoder by obtaining K outputs via BS and then using them to serve as expert supervision.

Our proposed algorithm, ∇BS can be trained in a stable manner using a mixture of the above strategies. MIXER (Ranzato et al., 2015), a hybrid strategy that uses both CE and RE trains the model for the first τ time steps using CE and then training with RE for the rest of the time; gradually reducing τ to 0. As proposed by Chen et al. (2018), QE can be used to train a reasonably good model that can be finetuned further using RE. Further, the *entire* model can be trained in an end-to-end fashion (denoted by EE) by backpropagating the gradients into the LSTM (the state transition function) producing the hidden states.

3. Related Work

Predicting Set-Valued Outputs There are comparatively few works that focus on predicting permutation invariant set-valued outputs using deep learning. Zaheer et al. (2017b) investigate commutative pooling operators for processing set-valued inputs, but with a focus on classification and regression problems. Rezatofighi et al. (2017) and Rezatofighi et al. (2018) predict set-valued outputs by learning both the cardinality and the state distribution of the target set. However, these approaches define the output space in terms of the possible subsets of some pre-existing support set, and so none of these approaches are applicable to the generative task of predicting *sets of sequences*. Recent work by Powers et al. (2018) aims at predicting diverse sequences but significantly differs from our work as they only consider the task of retrieval as opposed to sequence prediction.

Diverse Sequence Generation The most obvious way to generate sets of sequences is to apply beam search decoding to a standard neural sequence model such as an LSTM (Hochreiter & Schmidhuber, 1997). However, it is well known that the resulting sequences lack diversity (Gimpel et al., 2013; Li et al., 2015; Li & Jurafsky, 2016). A number of papers have tackled the problem of diverse sequence generation, either by modifying the training objective (Wang et al., 2017; Dai & Lin, 2017; Shetty et al., 2017), using model ensembles (Lee et al., 2016; He et al., 2018; Wang et al., 2016) or by modifying the decoding procedure (Vijayakumar et al., 2018; Li et al., 2016). However, none of these approaches are capable of directly learning the interactions between sequences in a set. Our model learns these interactions and can be optimized for any arbitrary set-level metric.

Trainable Sequence Decoding As detailed further in Section 2, our proposed approach constructs a set of K sequences in the output set incrementally, and can thus be interpreted as a trainable generalization of beam search. Therefore, although our motivations differ, our method is related to recent research that seeks to unify sequence model training and decoding regimes, either by modifying the training procedure (Andor et al., 2016; Wiseman & Rush, 2016; Goyal et al., 2017), or by casting sequence decoding as an optimization problem (Gu et al., 2017; Hoang et al., 2017; Chen et al., 2018). Notably, our approach differs from Goyal et al. (2017) as it avoids train-test mismatch by sampling in both phases and further, modeling intra-set interactions.

4. Experiments

In this section, we first discuss the trade-offs of different set-level metrics and then motivate a new set-level metric that evaluates the multi-modality in the output space. Next, we proceed to explaining the different evaluation metrics and baselines used in this work. We then report results on the visually-grounded language generation task of image captioning. Finally, we present a discussion on variants of our method and its applicability in different scenarios.

4.1. Set-Level Metrics for Language Generation.

Sequence level metrics for language generation tasks like BLEU (Papineni et al., 2002), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016) evaluate a decoded sequence by comparing it against a reference annotation in some feature representation[5]. Oracle accuracy, a popular set level metric (Zhang et al., 2006; Batra et al., 2012; Lee et al., 2016; Wang et al., 2017) used to evaluate the quality of decoded sets is constructed by using individual sequence level scores. Specifically, it corresponds to the best score achieved by any decoded sequence *i.e.* $\max_{\mathbf{y}\in\mathcal{Y}}\{\phi(\mathbf{y}|\mathbf{x})\}$. This serves the role of a downstream mechanism that selects the most appropriate sequence – for example, a human user or a reranker. While this is a reasonable choice when only one reference annotation is available, it is insufficient when multiple reference sentences are available; often the case with inherently multimodal tasks like image captioning. Oracle accuracy can be optimized by producing only one good caption that aligns well with the ground truth and therefore fails to penalize not *covering* the full variation present in human annotations.

To address this shortcoming of oracle accuracy, we propose a new set-level metric inspired by the classic facility location problem (Stollsteimer, 1963). Similar to oracle accuracy, the proposed *faccuracy* metric also uses sequence level scores and is given by $\sum_{\mathbf{r}\in\mathcal{R}}\max_{\mathbf{c}\in\mathcal{C}}\phi(\mathbf{r}|\mathbf{c})$ where $\mathcal{R}$ and $\mathcal{C}$ are the reference and candidate sequence sets respectively. This metric values output sets that contain sequences that maximize the sequence-level metric for each of the reference annotations; avoiding the shortcomings of oracle accuracy. Further, it is easy to notice that this set-level metric reduces to oracle accuracy when only one reference annotation is present. Additionally, the proposed faccuracy is also submodular; the notion of diminishing returns is observed as the value of adding new sequences after all references have been "covered" is little. In summary, the proposed fac metric extends oracle accuracy to consider higher order interactions between decoded sequences in a manner that naturally evaluates for the variation present in the reference set.

4.2. Baselines.

We are interested in the task of generating a set of captions that are highly valued by set-level metrics like faccuracy and oracle accuracy. All methods are evaluated and optimized (if applicable) using CIDEr (Vedantam et al., 2015) as the underlying sequence-level metric. Additionally, all methods decode $K = 5$ sequences per input.

We compare our approach against the most natural baseline – a standard sequence prediction model decoded using BS (which we denote by Seq-BS). Next, we compare against using a diversity-promoting decoding procedure, Diverse Beam Search (Vijayakumar et al., 2018) along with a sequence prediction model (Seq-DBS). Outperforming a tuned version of DBS implies that the our proposed algorithm introduces diversity appropriately without having to explicitly incentivize it. Additionally, we compare against Rennie et al. (2017), a sequence level model that uses REINFORCE to directly optimize the metric along with beam search decoding. This model, denoted by SCST, uses the score achieved by beam search under the current model as baseline to stabilize the training procedure. Further, we compare to the following ablations of our model that can be constructed by the various training strategies discussed in Section 2:

1. ∇BS-CE: This approach corresponds to training the model first using standard teacher forcing (CE, see Section 2). This approach is feasible when multiple annotations are available While this method suffers from both exposure bias and loss-evaluation mismatch, it still treats the outputs as a set and hence is capable of modeling intra-set interactions.

2. ∇BS-CE-EE: This is a natural extension of the previous baseline that backpropagates the gradient not only into the DSF but also into the underlying LSTM network. In practice, the finetuning begins after the DSF has been trained for a few rounds.

3. ∇BS-CE-RE: This approach corresponds to training the model first using CE and then using REINFORCE (RE);

[5] If $\mathcal{Y}_\mathbf{x}$ are the references corresponding to the input $\mathbf{x}$, we write $\phi(\cdot|\mathcal{Y}_\mathbf{x})$ for $\phi(\cdot|\mathbf{x})$ with some abuse of notation.

Dataset	Method	Faccuracy ($K = 5$)					Oracle Accuracy ($K = 5$)					distinct 4-grams
		BLEU	ROUGE	CIDEr	SPICE	METEOR	BLEU	ROUGE	CIDEr	SPICE	METEOR	
Flickr-8k	Seq-BS	0.2510	0.3149	1.7548	0.1534	0.1625	0.2712	0.4576	1.6564	0.1496	0.2351	17.38
	Seq-DBS	0.2583	0.3171	1.8374	0.1598	0.1607	0.2564	0.4535	1.6571	0.1572	0.2309	**64.44**
	SCST	0.2643	0.3204	1.8521	0.1632	0.1754	0.2644	0.4589	1.6792	0.1623	0.2386	25.79
	∇BS-CE	0.2681	0.3225	1.8946	0.1671	0.1751	0.2702	0.4592	1.7023	0.1643	0.2415	33.90
	∇BS-CE-EE	0.2685	0.3242	1.9142	0.1682	0.1751	0.2716	0.4597	1.7146	0.1645	0.2421	32.19
	∇BS-CE-RE	0.2707	0.3276	1.9238	0.1723	0.1822	0.2738	0.4618	1.7424	0.1664	0.2457	35.41
	∇BS-MIXER	0.2697	**0.3280**	1.9224	0.1782	0.1782	**0.2741**	0.4614	**1.7487**	0.1659	0.2462	35.85
	∇BS-CE-RE-EE	**0.2712**	0.3279	**1.9287**	**0.1806**	**0.1849**	0.2740	**0.4624**	1.7459	**0.1667**	**0.2464**	34.94
Flickr-30k	Seq-BS	0.2510	0.2916	1.7017	0.1643	0.1625	0.2781	0.4253	1.5850	0.1496	0.2351	18.04
	Seq-DBS	0.2625	0.2958	1.7726	0.1629	0.1607	0.2782	0.4292	1.5828	0.1572	0.2309	**64.18**
	SCST	0.2742	0.3124	1.7543	0.1664	0.1649	0.2804	0.4335	1.5974	0.1601	0.2390	27.42
	∇BS-CE	0.2788	0.3186	1.7724	0.1672	0.1653	0.2816	0.4378	1.6104	0.1617	0.2427	35.62
	∇BS-CE-EE	0.2793	0.3195	1.7812	0.1672	0.1657	0.2821	0.4467	1.6156	0.1621	0.2430	36.11
	∇BS-CE-RE	0.2794	0.3206	1.7942	0.1679	0.1665	0.2845	0.4514	1.6233	0.1627	0.2366	36.84
	∇BS-MIXER	**0.2798**	**0.3215**	1.8006	**0.1688**	0.1669	0.2839	**0.4529**	1.6229	0.1628	0.2471	35.91
	∇BS-CE-RE-EE	0.2794	0.3211	**1.8032**	0.1685	**0.1678**	**0.2846**	0.4519	**1.6238**	**0.1632**	**0.2472**	35.23
COCO	Seq-BS	0.2842	0.4892	1.5324	0.1724	0.2541	0.2839	0.5204	1.4208	0.1701	0.2570	20.04
	Seq-DBS	0.2915	0.4917	1.5266	0.1731	0.2585	0.2782	0.5247	1.4306	0.1708	0.2614	**68.18**
	SCST	0.2942	0.5012	1.5521	0.1739	0.2601	0.2804	0.5287	1.4421	0.1724	0.2664	30.42
	∇BS-CE	0.3015	0.5006	1.5721	0.1725	0.2605	0.2816	0.5276	1.4452	0.1722	0.2652	32.62
	∇BS-CE-EE	0.3011	0.5011	1.5784	0.1728	0.2609	0.2821	0.5288	1.4461	0.1726	0.2660	34.19
	∇BS-CE-RE	0.3056	0.5018	1.5894	0.1742	0.2656	0.2845	0.5296	1.4521	0.1759	0.2687	34.24
	∇BS-MIXER	0.3022	**0.5023**	1.5870	0.1736	0.2645	0.2839	0.5294	**1.4618**	0.1740	0.2689	33.01
	∇BS-CE-RE-EE	**0.3063**	0.5021	**1.5995**	**0.1745**	**0.2661**	**0.2846**	**0.5314**	1.4598	**0.1765**	**0.2697**	35.84

Table 1. On all the captioning datasets, ∇BS variants (MIXER and CE-RE-EE) outperform standard baselines and ablations. However, in terms of sheer diversity (as measured by distinct n-grams, Seq-DBS is still better. All the methods decode $K = 5$ outputs and further, we scale faccuracy values *in the table* by K for better readability.

optimizing directly for the set-level metric. Improving the model using RE "fixes" both loss-evaluation mismatch and exposure bias.

4. ∇BS-MIXER (Ranzato et al., 2015): This approach is similar to ∇BS-CE-RE but differs in that the two methods operate simultaneously instead of being applied one after the other. The approach works by using CE for the first $\tau \in [T]$ steps and then trains via RE for the rest $T - \tau$ steps. The value of τ is gradually reduced from T (corresponding to ∇BS-CE) to 0 (corresponding to ∇BS-RE) thereby following a curriculum that spans a spectrum of training methods.

4.3. Image Captioning

In this section, we explain the experimental setup and report results for the image-captioning task.

Datasets and Models. We show results on three captioning datasets of increasing size – Flickr8k, Flickr30k (Young et al., 2014) and the large scale COCO dataset (Lin et al., 2014). All of these datasets are multimodal and have 5 captions associated with each image. For the first two Flickr datasets, 1000 images each are used for validation and testing while using the rest (6000 and ∼28000 respectively) for training. For COCO, a similar split is used but the number of images used for validation and testing each is 5000.

The underlying sequence level model is an encoder-decoder architecure proposed by Vinyals et al. (2015); a single layer LSTM with 1024 hidden units. For the DSF, we use a two-layer MLP, as defined in (9) with $d = 1024$ (LSTM hidden size) and $m = 512$. The input image is treated as the first word and is represented using activations of the penultimate layer of ResNet-152 (He et al., 2016) network, pretrained on Imagenet (Deng et al., 2009). Both the DSF and the LSTM (in the case of EE) are trained using Adam (Kingma & Ba, 2014) with a learning rate of $1e - 4$ and $1e - 5$ respectively. We set the beam size $K = 5$ in all our experiments. As mentioned in Section 2, we first do a coarse selection using a standard sequence model; inputting only the top-100 alternatives corresponding to each partial solution to the DSF. Importantly, note that this trick is required *only* to speed up the training phase. Further, all variants of o ur approach are warm started from standard sequence prediction trained via MLE.

Evaluation. When training using RE, we use CIDEr (Vedantam et al., 2015) to compute faccuracy and optimize for it; TF-IDF vectors for computing CIDEr are obtained from COCO-validation split. However, we report results on all the commonly used captioning metrics – BLEU-4 (Papineni et al., 2002), METEOR (Denkowski & Lavie, 2014), ROUGE (Lin, 2004), CIDEr and SPICE (Anderson et al., 2016). Additionally, we report *distinct n-grams*, a

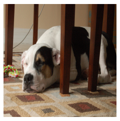
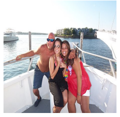
	Captions produced by			
	Humans	Beam Search	Diverse Beam Search	∇BS-CE-RE
	A white and brown dog is asleep underneath a small table A dog is sleeping under a table A dog is sleeping under a chair A spotted dog is asleep under a table A dog is sleeping on the floor	A black and white dog is looking at the camera A black and white dog sitting on the floor A black and white dog laying on the floor A black and white dog laying on the floor A brown and white dog sitting in the room	A dog is looking out of a window A dog sitting in a window looking at something A black and white dog is sitting on the floor The dog is sitting on the floor looking at the camera A white dog is looking at a cat	A white dog sitting on the ground A black and white dog sitting on the floor A black and white dog laying on the floor The dog is sitting in the room The dog is laying on the floor
	Three people posing on a boat Two women and a man on a boat are posing for a picture Three people are riding in a boat on a sunny day A group of people on a boat Three friends enjoying a boat ride	A man and a woman on a boat in the water A man and a woman standing next to a small boat A man and a woman sitting on a boat in the water A man standing next to a woman on a boat A man and a woman riding a boat	A man and a woman are standing on a boat A man and a woman standing on a boat with a dog Two people standing on a boat in the water Two people standing in front of a boat The young man is holding his cellphone	A man and a woman standing in front of a boat Two people standing on a boat in the water A woman in pink dress standing on a boat A young man holding his cellphone An older man and a woman standing on a boat

Table 2. Captions produced by our approach ∇BS compared against Human annotations, BS and DBS for two images – one `simple` image that has less variation in human annotations and a `complex` image that has multiple objects and interactions, exhibiting greater variation in human generated captions. While BS tends to be largely repetitive, DBS, with parameters tuned based on a validation set, tends to produce diverse captions while some of which might not be applicable to the image. On the other hand, ∇BS strikes a balance between the two procedures in terms of diversity, aligning with the observations from Table 1.

Split	Faccuracy	
	CIDEr	SPICE
`simple`	1.5723	0.1724
`average`	1.6015	0.1751
`complex`	1.6012	0.1754

Table 3. The ∇BS-CE-RE-EE variant of our model performs equally well (score on the entire COCO test split is 1.5995 and 0.1745 for CIDEr and SPICE respectively) across all levels of complexity; demonstrating that learning to decode learns to promote diversity while being aware of the contents of the image.

metric introduced by (Li & Jurafsky, 2016) to serve as an indicator of the diversity in the decoded lists. Specifically, we report the number of unique 4-grams and normalize it by the number of words to bias against larger sequences.

As we see from Table 1, variants of ∇BS outperform standard sequence models used with BS or DBS. Further, they also outperform (Rennie et al., 2017) that directly optimizes for the metric. Among the proposed decoders, the ∇BS-MIXER and ∇BS-CE-RE-EE variants perform the best, each performing best on certain metrics. Importantly, these trends hold across all three data-sets used in this experiment.

4.4. Discussion.

In this subsection, we discuss different variants of our method and its applicability in different scenarios.

Is Diversity Always Required? While diversity in the decoded captions is beneficial, it may not always be necessary. For example, it may not be possible to describe an image containing one object (*e.g.* a close up of a cat) in diverse ways. Following the analysis of (Vijayakumar et al., 2018), we divide the images in the test set into three sets – `simple`, `average` and `complex`, based on their image complexity scores (Ionescu et al., 2016). These scores are higher for images with many objects and are in some sense, reflective of the "complexity" of the image. As seen from Table 2 and 3, our method ∇BS-CE-RE-EE performs consistently well on all three splits of varying complexity.

Set-metrics with Combinatorial Constraints. To demonstrate the ability of our method in handling arbitrary set-level constraints, we optimize set-level metrics with combinatorial constraints; for *e.g.* in the context of automatic response suggestion (Kannan et al., 2016), such a set-level metric can reward the first two sequences based on the presence of positive sentiment and the rest, on negative sentiment. In our image captioning setup, we instantiate such a metric by using CIDEr to reward the first two sequences and SPICE for the remaining three ($K = 5$). We observe that first 2 sequences get a higher CIDEr score (an average of 1.1623 against a SPICE score of 0.1622) and similarly, the remaining three sequences achieve a higher SPICE score (0.1698 as compared to a CIDEr score of 1.0624).

5. Conclusion

Producing a set of K outputs is beneficial for tasks that are inherently multimodal, admitting multiple correct outputs for a single input. Further, many tasks that desire a single best output produce such a set of outputs as an intermediate step. Despite its widespread usage, existing sequence prediction models used in conjuction with decoding strategies like BS fail to produce good output sets; often producing largely redundant sequences with minor varations. To address this we propose ∇BS, a trainable decoder for sets of sequences. Our method accounts for higher order interactions like diversity by modeling intra-set interactions and can be tuned to optimize arbitrary set-level metrics. Finally, we report results on the language generation task of image-captioning and include a discussion of variants of our method and its applicability in different scenarios.

References

Anderson, P., Fernando, B., Johnson, M., and Gould, S. Spice: Semantic propositional image caption evaluation. 2016.

Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., and Collins, M. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*, 2016.

Batra, D., Yadollahpour, P., Guzman-Rivera, A., and Shakhnarovich, G. Diverse M-Best Solutions in Markov Random Fields. 2012.

Bilmes, J. and Bai, W. Deep submodular functions. *arXiv preprint arXiv:1701.08939*, 2017.

Chen, X., Hao Fang, T.-Y. L., Vedantam, R., Gupta, S., Dollar, P., and Zitnick, C. L. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*, 2015.

Chen, Y., Li, V. O., Cho, K., and Bowman, S. R. A stable and effective learning strategy for trainable greedy decoding. *arXiv preprint arXiv:1804.07915*, 2018.

Dai, B. and Lin, D. Towards diverse and natural image descriptions via a conditional gan. 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009.

Denkowski, M. and Lavie, A. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of The Ninth Workshop on Statistical Machine Translation*, 2014.

Gimpel, K., Batra, D., Dyer, C., and Shakhnarovich, G. A systematic exploration of diversity in machine translation. 2013.

Goyal, K., Neubig, G., Dyer, C., and Berg-Kirkpatrick, T. A continuous relaxation of beam search for end-to-end training of neural sequence models. *arXiv preprint arXiv:1708.00111*, 2017.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov): 1471–1530, 2004.

Gu, J., Cho, K., and Li, V. O. Trainable greedy decoding for neural machine translation. *arXiv preprint arXiv:1702.02429*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, X., Haffari, G., and Norouzi, M. Sequence to sequence mixture model for diverse machine translation. *arXiv preprint arXiv:1810.07391*, 2018.

Hoang, C. D. V., Haffari, G., and Cohn, T. Towards decoding as continuous optimisation in neural machine translation. In *EMNLP*, 2017.

Hochreiter, S. and Schmidhuber, J. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pp. 473–479, 1997.

Hong, K., Conroy, J. M., Favre, B., Kulesza, A., Lin, H., and Nenkova, A. A repository of state of the art and competitive baseline summaries for generic news summarization. In *LREC*, pp. 1608–1616, 2014.

Ionescu, R. T., Alexe, B., Leordeanu, M., Popescu, M., Papadopoulos, D., and Ferrari, V. How hard can it be? Estimating the difficulty of visual search in an image. 2016.

Jiang, S. and de Rijke, M. Why are sequence-to-sequence models so dull? understanding the low-diversity problem of chatbots. *arXiv preprint arXiv:1809.01941*, 2018.

Kalyan, A., Lee, S., Kannan, A., and Batra, D. Learn from your neighbor: Learning multi-modal mappings from sparse annotations. 2018.

Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., Corrado, G., Lukács, L., Ganea, M., Young, P., et al. Smart reply: Automated reponse suggestion for email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lee, S., Prakash, S. P. S., Cogswell, M., Ranjan, V., Crandall, D., and Batra, D. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 2119–2127, 2016.

Li, J. and Jurafsky, D. Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*, 2016.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. A diversity-promoting objective function for neural conversation models. 2015.

Li, J., Monroe, W., and Jurafsky, D. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*, 2016.

Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

Lin, H. and Bilmes, J. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 510–520. Association for Computational Linguistics, 2011.

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

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functionsi. *Mathematical programming*, 14(1):265–294, 1978.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. 2002.

Powers, T., Fakoor, R., Shakeri, S., Sethy, A., Kainth, A., Mohamed, A.-r., and Sarikaya, R. Differentiable greedy networks. *arXiv preprint arXiv:1810.12464*, 2018.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. Self-critical sequence training for image captioning. In *CVPR*, volume 1, pp. 3, 2017.

Rezatofighi, S. H., Kumar B G, V., Milan, A., Abbasnejad, E., Dick, A., and Reid, I. Deepsetnet: Predicting sets with deep neural networks. In *ICCV*, 2017.

Rezatofighi, S. H., Milan, A., Shi, Q., Dick, A., and Reid, I. Joint learning of set cardinality and state distribution. In *AAAI*, 2018.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Shen, L., Sarkar, A., and Och, F. J. Discriminative reranking for machine translation. In *North American Association for Computational Linguistics (NAACL)*, pp. 177–184, 2004.

Shetty, R., Rohrbach, M., Hendricks, L. A., Fritz, M., and Schiele, B. Speaking the same language: Matching machine to human captions by adversarial training. In *ICCV*, 2017.

Stollsteimer, J. F. A working model for plant numbers and locations. *Journal of Farm Economics*, 45(3):631–645, 1963.

Tschiatschek, S., Sahin, A., and Krause, A. Differentiable submodular maximization. *arXiv preprint arXiv:1803.01785*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Vedantam, R., Lawrence Zitnick, C., and Parikh, D. Cider: Consensus-based image description evaluation. 2015.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D. J., and Batra, D. Diverse beam search: Decoding diverse solutions from neural sequence models. In *AAAI*, 2018.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and tell: A neural image caption generator. 2015.

Wang, L., Schwing, A., and Lazebnik, S. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *Advances in Neural Information Processing Systems*, pp. 5758–5768, 2017.

Wang, Z., Wu, F., Lu, W., Xiao, J., Li, X., Zhang, Z., and Zhuang, Y. Diverse image captioning via grouptalk. In *IJCAI*, pp. 2957–2964, 2016.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Wiseman, S. and Rush, A. M. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*, 2016.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. 2014.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., and Smola, A. Deep sets. arxiv preprint. *arXiv preprint arXiv:1703.06114*, 7, 2017a.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In *NIPS*, 2017b.

Zhang, Y., Hildebrand, A. S., and Vogel, S. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 216–223. Association for Computational Linguistics, 2006.