The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Diverse Beam Search for Improved Description of Complex Scenes

Ashwin K Vijayakumar,[1] **Michael Cogswell,**[1] **Ramprasaath R Selvaraju,**[1]
Qing Sun,[2] **Stefan Lee,**[1] **David Crandall,**[3] **Dhruv Batra**[1,4]
[1]Georgia Tech, [2]Virginia Tech, [3]Indiana University, [4]Facebook AI Research
{ashwinkv, cogswell, ramprs, steflee, dbatra}@gatech.edu
sunqing@vt.edu, djcran@indiana.edu

Abstract

A single image captures the appearance and position of multiple entities in a scene as well as their complex interactions. As a consequence, natural language grounded in visual contexts tends to be diverse – with utterances differing as focus shifts to specific objects, interactions, or levels of detail. Recently, neural sequence models such as RNNs and LSTMs have been employed to produce visually-grounded language. Beam Search, the standard work-horse for decoding sequences from these models, is an approximate inference algorithm that decodes the top-B sequences in a greedy left-to-right fashion. In practice, the resulting sequences are often minor rewordings of a common utterance, failing to capture the multimodal nature of source images. To address this shortcoming, we propose Diverse Beam Search (DBS), a diversity promoting alternative to BS for approximate inference. DBS produces sequences that are significantly different from each other by incorporating diversity constraints within groups of candidate sequences during decoding; moreover, it achieves this with minimal computational or memory overhead. We demonstrate that our method improves both diversity and quality of decoded sequences over existing techniques on two visually-grounded language generation tasks – image captioning and visual question generation – particularly on complex scenes containing diverse visual content. We also show similar improvements at language-only machine translation tasks, highlighting the generality of our approach.

1 Introduction

A picture is often said to be worth a thousand words, owing this high valuation to its capability to simultaneously capture multiple objects and their interactions precisely. Communicating this rich information in natural language requires providing many details about the scene at varying levels of granularity, resulting in a great deal of diversity in visually-grounded language. Recently, automated approaches for generating visually-grounded language based on neural sequence models have been studied (Vinyals et al. 2015; Venugopalan et al. 2015; Mostafazadeh et al. 2016; Das et al. 2017); however, in practice, utterances generated from these models often tend to be generic and fail to recover the diversity observed in human annotations.

Modeling Visually-Grounded Language. Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), or more generally, neural sequence models have been extensively used for modeling time-series in a data-driven manner – including, standard sequence-to-sequence problems such as speech recognition (Graves, Mohamed, and Hinton 2013), machine translation (Bahdanau, Cho, and Bengio 2014), and conversation modeling (Vinyals and Le 2015). More recently, neural sequence models have been applied to visually-grounded language generation tasks like image and video captioning (Vinyals et al. 2015; Venugopalan et al. 2015), question generation (Mostafazadeh et al. 2016), and dialog (Das et al. 2017). In these tasks, neural sequence models are typically trained to estimate the likelihood of a sequence of output tokens $\mathbf{y} = (y_1, \ldots, y_T)$ from a finite vocabulary $\mathcal{V}$, conditioned on some input $\mathbf{x}$. For example, in image captioning, the input $\mathbf{x}$ is a continuous representation of a source image as encoded by a Convolutional Neural Network (CNN) and the output $\mathbf{y}$ is a natural language description of the scene depicted in the source image.

Inference in RNNs. At test time, Maximum a Posteriori (MAP) inference must be performed to decode the most likely sequence given an input image. However, the space of all T length sequences consists of $|\mathcal{V}|^T$ possibilities; therefore, exact inference is intractable even for modestly sized tasks. Instead, approximate inference algorithms like Beam Search (BS) are commonly used to decode likely sequences.

BS is a heuristic graph-search algorithm that maintains the B most-likely partial sequences expanded in a greedy left-to-right fashion (Fig. 1 (middle) shows a sample search tree). Despite its widespread usage, it is generally known to produce generic or "safe" outputs. For example, generic captions like "Animals standing in the field" or responses such as "I can't tell" in dialog are applicable to a wide range of images and hence, are largely uninformative. Equally problematic, the top-B outputs from BS lack diversity and typically express an identical sentiment through minor rewordings (often only in the last few words). While this behavior is disadvantageous for many reasons (including being computationally wasteful), we argue that the most adverse effects occur in cases where $\Pr(\mathbf{y}|\mathbf{x})$ truly is multimodal; as is often the case in language generation tasks where there is generally not a single 'correct' utterance.

Fig. 1 highlights these deficiencies in an example image

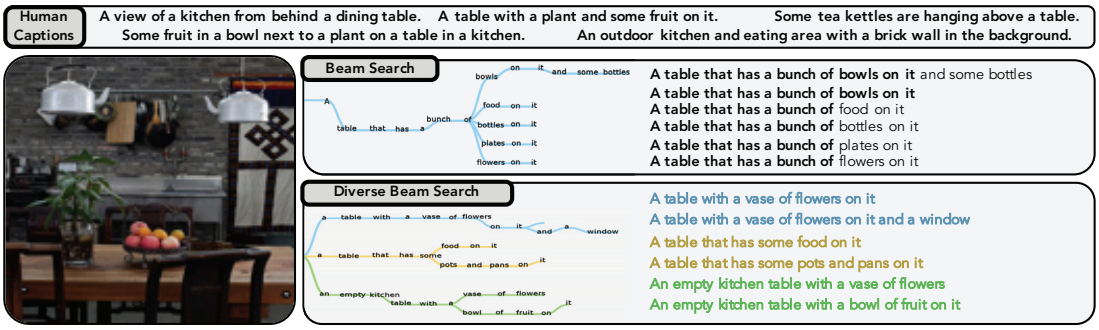

Figure 1: Comparing image captioning outputs decoded by BS (top) and our method, Diverse Beam Search (middle) – we notice that BS captions are near-duplicates with similar shared paths in the search tree and minor variations in the end. In contrast, DBS captions are significantly diverse and similar to the variability in human-generated ground truth captions (bottom).

captioning task. The human captions (top) show a range of phrasings and focus on different objects (*table, plant, fruit, kettles*), relationships (*on, in, above*) and granularity (*kitchen vs. objects in the kitchen*). The BS based captions (middle-top) in contrast are generic captions that complete a single root sentence with various objects typically found on a table (*bowls, food, bottles, plates, flowers*), though many of them are not present on *this* table. It is clear that producing B nearly identical, generic captions is woefully inadequate to reflect the space of *relevant* descriptions.

Overview and Contributions. To address this shortcoming, we propose *Diverse Beam Search (DBS)* – a general framework for decoding a set of diverse sequences that can be used as an *alternative* to BS. At a high level, DBS decodes diverse lists by dividing candidate solutions into groups and enforcing diversity between groups. DBS decoded captions in Fig. 1 (bottom) show higher variability in phrasing and focus more on objects actually in the scene. Drawing from work in the probabilistic graphical models literature on Diverse M-Best (DivMBest) MAP inference (Batra et al. 2012; Prasad, Jegelka, and Batra 2014; Kirillov et al. 2015), we optimize an objective comprised of two terms – the sequence likelihood under the neural sequence model and a dissimilarity term that encourages sequence across groups to differ. This diversity-augmented model score is optimized in a *doubly greedy* manner – greedily maximizing both along time (like BS) and across groups (like DivMBest).

We report results on two visually grounded tasks – image captioning and visual question generation and machine translation. Our experiments show that DBS consistently outperforms baseline methods in terms of both diversity-related and task-specific quality metrics. Moreover, we find that both these improvements and human preference for DBS decoded outputs increase on tasks grounded in more complex images (*i.e.* those *requiring* a greater deal of diversity). We also show improvements over BS on non-visual machine translation tasks. Overall, our algorithm decodes high-quality, diverse sequence sets while being simple to implement and comparable to BS in terms of computation and memory requirements. To aid transparency and reproducibility, our code for DBS is available at https://github.com/ashwinkalyan/dbs. A demo of our method is available at http://dbs.cloudcv.org/.

2 Related Work

Diverse M-Best Lists. The task of generating diverse structured outputs from probabilistic models has been studied extensively (Kirillov et al. 2016; Batra et al. 2012; Kirillov et al. 2015; Prasad, Jegelka, and Batra 2014). Batra et al. (2012) formalized this task for Markov Random Fields as the DivMBest problem and presented a greedy approach which solves for outputs iteratively, conditioning on previous solutions to induce diversity. Kirillov et al. (2015) show how these solutions can be found jointly for certain kinds of energy functions; however, these techniques are not directly applicable to decoding from RNNs.

Most related to our proposed approach is the work of Gimpel et al. (2013), who apply DivMBest to machine translation using beam search as a black-box inference algorithm. Specifically, in this approach, DivMBest knows nothing about the inner-workings of BS and simply makes M sequential calls to BS to generate M diverse solutions. This approach is rather wasteful because BS is run from scratch every time and although each call to BS produces B solutions, only *one solution* is retained by DivMBest. In contrast, the approach developed in this paper (DBS) avoids these shortcomings by integrating diversity within BS such that *no beams are wasted*. By running multiple beam searches in parallel and at staggered time offsets, we obtain large time savings, making our method comparable to *a single run* of classical BS and *M times faster* than (Gimpel et al. 2013). One potential disadvantage of our method with respect to (Gimpel et al. 2013) is that sentence-level diversity metrics cannot be incorporated in DBS as diversity is encouraged amongst groups before waiting for them to completely decode a sentence. However, as observed empirically by us and (Li et al. 2015), initial words tend to disproportionately impact the diversity of the resulting sequences – suggesting that later words may not be important for inducing diversity.

Diverse Decoding for RNNs. Efforts have been made by Li et al. (2015) and Li and Jurafsky (2016) to produce diverse decodings from recurrent models for conversation model-

ing and machine translation by introducing novel heuristics within the Beam Search (BS) algorithm.

Li and Jurafsky (2016) proposes a BS diversification heuristic that discourages beams from sharing common roots, implicitly resulting in diverse lists. Introducing diversity through a formal objective (as in DBS) rather than via a procedural heuristic provides the flexibility to incorporate different notions of diversity and control the exploration-exploitation trade-off. Furthermore, we find that DBS significantly outperforms this approach in our experiments on multiple datasets. Li et al. (2015) introduce a novel decoding objective that maximizes mutual information between inputs and predictions to penalize generic sequences. The goal is to penalize utterances that occur frequently (*i.e*. generic decodings) rather than penalizing similarity between generated sequences – which in principle is *complementary* to both DBS and (Li and Jurafsky 2016). Furthermore, evaluating the 'genericness' of a sequence *requires training a new input-independent language model* for the target language while DBS just requires a measure of diversity between sequences. Combining these complementary techniques is left as interesting future work.

Sequence Optimization. In an orthogonal line of work, (Wiseman and Rush 2016) directly learn to search in the exponential output space to fix the shortcomings of using seq2seq models. They integrate both the seq2seq architecture and the search problem of finding the top-sequence via optimizing for both the negative log-likelihood and search-based losses to obtain significant improvements over the standard training and inference pipeline. In contrast, our approach is an *inference-only* technique that does not require any re-training that works in a model-agnostic fashion.

3 Preliminaries: RNNs, Beam Search, and DivMBest

We begin with a refresher on Beam Search for inference in RNNs and DivMBest before detailing our approach. For notational convenience, we denote the set of natural numbers from 1 to n with $[n]$ and use $\mathbf{v}_{[n]} = [v_1, \dots, v_n]^\intercal$ to index the first n elements of a vector $\mathbf{v} \in \mathbb{R}^m$.

RNNs are neural sequence models trained to estimate the likelihood of sequences of tokens from a finite dictionary $\mathcal{V}$ given an input $\mathbf{x}$. The RNN updates its internal state and estimates the conditional probability distribution over the next output given the input and all previous output tokens, $\log \Pr(y_t|\mathbf{y}_{[t-1]}, \mathbf{x})$. We write the log probability of a sequence $\mathbf{y} \in \mathcal{V}^T$ of length T as $\Theta_T(\mathbf{y}; \mathbf{x}) = \sum_{t\in[T]} \log \Pr(y_t|\mathbf{y}_{[t-1]}, \mathbf{x})$. The decoding problem is then the task of finding a sequence $\mathbf{y}$ that maximizes $\Theta_T(\mathbf{y}; \mathbf{x})$.

As each output is conditioned on all the previous outputs, decoding the optimal length-T sequence in this setting can be cast as MAP inference on a T-order Markov chain with nodes corresponding to output tokens at each time step. Not only does the size of the largest factor in such a graph grow as $|\mathcal{V}|^T$, but computing these factors also requires repetitively evaluating the sequence model. Thus, approximate inference algorithms are employed, with the most prevalent method being Beam Search (BS).

Beam Search is a heuristic search algorithm which stores the top-B highest scoring partial solutions at each time step; where B is known as the *beam width*. At time t, BS considers all possible single token extensions of existing beams and retains the B highest scoring extensions.

Let us denote the set of B solutions held by BS at the end of time $t-1$ as $Y_{[t-1]} = \{\mathbf{y}_{1,[t-1]}, ..., \mathbf{y}_{B,[t-1]}\}$. At each time step, BS considers all possible single token extensions of these beams given by the set $\mathcal{Y}_t = \{\mathbf{y} \mid \mathbf{y}_{[t-1]} \in Y_{[t-1]} \wedge y_t \in \mathcal{V}\}$ and retains the B highest scoring extensions. More formally, at each step the beams are updated as

$$Y_{[t]} = \underbrace{\operatorname*{argmax}_{y_1,\dots,y_B \in \mathcal{Y}_t}}_{\text{pick top-B}} \sum_{b\in[B]} \Theta_t(\mathbf{y}_{b,[t]}; \mathbf{x}) \quad (1)$$
$$s.t. \quad \underbrace{\mathbf{y}_i \neq \mathbf{y}_j}_{\text{non-identical beams}} \quad \forall\, i \neq j \text{ and } i, j \in [B].$$

The above objective can be trivially maximized by sorting all $B\times|\mathcal{V}|$ members of $\mathcal{Y}_t$ by their log probabilities and selecting the top B. This process is repeated until time T and the complete beams are sorted by log probabilities.

While this method allows for multiple sequences to be explored in parallel, most completions tend to stem from a single highly valued beam (Li and Jurafsky 2016)– resulting in outputs that are often only minor perturbations of a single sequence. To make the decoded lists reflect the variation present in human-generated language, we show how the beam search objective can be augmented to include a diversity constraint.

DivMBest. Batra et al. (2012) formalize the task of generating M diverse but likely solutions as the DivMBest problem and develop a greedy incremental approach which solves for one solution at a time conditioned on the previous ones.

Let $S(\mathbf{y}; \mathbf{x})$ measure the quality of a solution $\mathbf{y} \in \mathcal{Y}$ and $\Delta(\cdot, \cdot)$ measure dissimilarity between elements of $\mathcal{Y}$. In this greedy approach, solutions are found sequentially through a dissimilarity-constrained maximization with respect to previous solutions,

$$\mathbf{y}^m = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} S(\mathbf{y}; x) \;\; s.t. \;\; \Delta(\mathbf{y}, \mathbf{y}^i) \geq k_i \;\; \forall i < m \quad (2)$$

which enforces that new solutions must be sufficiently far from existing ones by factors $\boldsymbol{k}=\{k_i|i\in[m-1]\}$.

In general, this problem is NP-hard and Batra *et al*. instead formulate the Lagrangian relaxation of this objective,

$$g(\boldsymbol{\lambda}) = \max_{\mathbf{y}\in\mathcal{Y}} \; S(\mathbf{y}; x) + \sum_{i=1}^{m-1} \lambda_i(\Delta(\mathbf{y}, \mathbf{y}^i) - k_i), \quad (3)$$

where $\boldsymbol{\lambda} = \{\lambda_i | i \in [m-1]\}$ is the set of Lagrange multipliers which scale the cost of violating each constraint. In practice, setting distance limits $\boldsymbol{k}$ is unintuitive; however, the authors note that tuning $\boldsymbol{\lambda}$ directly is analogous to maximizing $g(\cdot)$ with respect to $\boldsymbol{\lambda}$ for some unknown set of limits and represents a more intuitive linear trade-off between quality and dissimilarity of solutions.

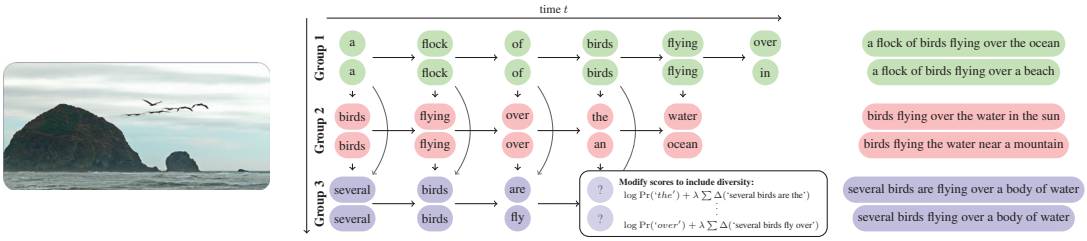

Figure 2: Diverse beam search operates left-to-right through time and top to bottom through groups. Diversity between groups is combined with joint log probabilities, allowing diverse continuations to be found efficiently.

With fixed values of $\boldsymbol{\lambda}$ and prior solutions $\mathbf{y}^1, \ldots, \mathbf{y}^{m-1}$, the inner maximization over $\mathcal{Y}$ inside $g(\cdot)$ is a function only of $\mathbf{y}$. Given an algorithm capable of maximizing the original $S(\mathbf{y}; x)$, the next diverse solution can be found by applying the same approach on the diversity-augmented criteria $S_\Delta(\mathbf{y}; x) = S(\mathbf{y}; x) + \sum_{i=1}^{m1} \lambda_i \Delta(\mathbf{y}, \mathbf{y}^i)$.

Gimpel et al. (2013) apply DivMBest to machine translation by using beam search to maximize this objective, generating M diverse solutions by performing M complete beam searches (with B beams), keeping the highest ranked solution from each, and discarding the remaining $B-1$ sequences each time. The root cause of this inefficiency is the treatment of BS as a black-box optimizer and the implementation of DivMBest as a naïve outer-for-loop around it. In the next section, we present Diverse Beam Search, which directly incorporates *diversity within beam search itself* to improve diversity without incurring this expense.

4 Approach: Diverse Beam Search (DBS)

In this section, we present Diverse Beam Search, an algorithm that tightly integrates diversity *within the BS* search process to efficiently produce diverse sequences.

Overview and Intuitions. To induce diversity in the selection of beam completions during beam search, we consider augmenting the objective in Equation 1 with a DivMBest style dissimilarity term, $\lambda\Delta(\cdot)$. This formulation would encourage all beams to differ from one another, with each *seeking out a different mode* of the output distribution. However, BS is greedy through time and a single beam may be insufficient to find highly-likely sequences from each mode, so we further propose dividing the set of beams into groups and encouraging diversity only between groups and not within. By dividing our beam budget in this way, we can vary the number of groups to balance between exploration of the space (more groups with fewer beams) and exploitation of local maximum (fewer groups with more beams).

Figure 2 displays a snapshot of the proposed method on an image captioning task with G=3 groups comprised of B'=2 beams each. Each group can be viewed as a smaller, independent beam search operating under a diversity augmented objective based on previous groups' search paths. As each group must wait for the prior groups to be processed at each time step, groups are extended forward in time along a staggered beam-front. In the graphic, the third group is being stepped forward at time step $t = 4$ and the previous groups have already been extended for this time step. In this example, we use hamming distance to measure diversity which rewards using different words from those used by previous groups at the same time step – 'birds', 'the', and 'an' in the example. After the diversity-augmented log-probabilities are computed like in DivMBest, the top B' extensions for the third group can be found by a standard beam search step. Thus, our approach is *doubly greedy* – both along *time* like BS and across *groups* like DivMBest. Specifically, the algorithm proceeds in a 'column-major' fashion, greedily optimizing all the groups at each time step. We now detail our approach which we refer to as Diverse Beam Search (DBS).

DBS Formulation. More formally, consider a partition of the beams, $Y_{[t]}$, into G groups $Y^g_{[t]}, g{\in}[G]$ each containing $B'{=}B/G$ beams (a non-uniform beam distribution is possible in practice). At each time step t, we greedily update each group g by selecting extensions of currently held partial solutions $Y^g_{[t]}{=}\{\mathbf{y}^g_{1,[t]}, \ldots, \mathbf{y}^g_{B',[t]}\}$ that maximize a linear combination of sequence likelihood and diversity with respect to previous groups, similar to DivMBest.

We begin by defining a diversity function $\Delta(\mathbf{y}_{[t]}, Y^g_{[t]})$ which measures the *dissimilarity* between a sequence $\mathbf{y}_{[t]}$ and group $Y^g_{[t]}$. While $\Delta(\cdot,\cdot)$ can take many forms, for simplicity we define one broad class that decomposes across beams within a group. We write the general form as

$$\Delta\left(\mathbf{y}_{[t]}, Y^g_{[t]}\right) = \overbrace{\sum_{b=1}^{B'} \underbrace{\delta\left(\mathbf{y}_{[t]}, \mathbf{y}^g_{b,[t]}\right)}_{\text{dissimilarity}}}^{\text{sum over all previous group beams}} \tag{4}$$

where $\delta(\cdot,\cdot)$ is a measure of sequence dissimilarity – *e.g.* a negative cost for each co-occurring n-gram in two sentences or distance between distributed sentence representations.

In analogy to DivMBest approaches, we optimize each group while holding previously extended groups fixed, incorporating the diversity term $\Delta(\cdot,\cdot)$ into the BS objective presented in (1). For time step t, we can write this diversity-augmented optimization for updating group g as

$$Y_t^g = \underbrace{\underset{\mathbf{y}_1^g,\ldots,\mathbf{y}_{B'}^g}{\operatorname{argmax}}}_{\text{select top-}B} \sum_{b\in[B']} \underbrace{\Theta_t\left(\mathbf{y}_{b,[t]}^g\right)}_{\text{score of extensions}} + \underbrace{\lambda \sum_{h=1}^{g-1} \Delta\left(\mathbf{y}_{b,[t]}^g, Y_{[t]}^h\right)}_{\text{diversity w.r.t. previous groups}} \quad (5)$$

$$s.t.\ \lambda \geq 0,\ \underbrace{\mathbf{y}_{i,[t]}^g \neq \mathbf{y}_{j,[t]}^g \forall i \neq j}_{\text{non-identical extensions}}$$

This modified objective is a trade-off between the likelihood of the completions and their diversity with respect to previously extended groups. As the previous groups are held fixed, Eq. 5 is only a function of the possible extensions. As such, the log-probabilities of the completions can be augmented with the diversity term – reducing this problem to a standard BS step with can be solved by sorting the extension scores. We repeat this for each group at each time step.

Our approach is formalized in Alg. 1 and consists of two main steps performed for each group at each time step –

1) augmenting the log probabilities of all possible extensions with the diversity term computed from previously advanced groups (Algorithm 1, Line 7) and,
2) running one step of a smaller BS with B' beams using the augmented log probabilities to select extensions for the current group (Algorithm 1, Line 9).

After all sequences have been extended to a preset max length or otherwise terminated, all solutions from each group are combined and sorted by log probability.

There are a number of advantages worth noting about this approach. By encouraging diversity between beams at each step (rather than just between highest ranked solutions like in (Gimpel et al. 2013)), our approach rewards each group for spending its beam budget to explore different parts of the output space rather than repeatedly chasing sub-optimal beams from prior groups. Furthermore, the time-staggered group structure enables each group beam search to be performed in parallel with a time offset. This parallel algorithm completes in $T + G$ time steps compared to $T * G$ running time for a black-box approach of Gimpel *et al*. (Gimpel et al. 2013). Finally, we note that as the first group is not conditioned on other groups, DBS is guaranteed to perform at least as well as a beam search of size B'.

4.1 Analysis of Hyper-parameters

Here we discuss the impact of the number of groups, strength of diversity, and various forms of diversity for language models. Note that the parameters of DBS (and other baselines) were tuned on a held-out validation set for each experiment. Further discussion and full experimental results are detailed in the supplement.

Number of Groups (G). Setting $G{=}B$ allows for the maximum exploration of the search space, while setting $G{=}1$ reduces DBS to BS, resulting in increased exploitation of the search-space around the 1-best decoding. Empirically, we find that maximum exploration correlates with improved oracle accuracy and hence use $G{=}B$ to report results.

Diversity Strength (λ). The diversity strength λ specifies the trade-off between the model score and diversity terms.

Algorithm 1: Diverse Beam Search

1 Diverse Beam Search with G groups using B beams
2 **for** $t = 1, \ldots T$ **do**
  // perform one step of beam search
3 $Y_{[t]}^1 \leftarrow \operatorname{argmax}_{(\mathbf{y}_{1,[t]}^1,\ldots,\mathbf{y}_{B',[t]}^1)} \sum_{b\in[B']} \Theta_t(\mathbf{y}_{b,[t]}^1)$
4 s.t. $\mathbf{y}_{i,[t]}^1 \neq \mathbf{y}_{j,[t]}^1 \ \forall i \neq j$
5 **for** $g = 2, \ldots G$ **do**
    // augment log probabilities
6 $\Theta_t(\mathbf{y}_{b,[t]}^g) \leftarrow$ $\Theta_t(\mathbf{y}_{b,[t]}^g) + \lambda \sum_{h=1}^{g-1} \Delta(\mathbf{y}_{b,[t]}^g, Y_{[t]}^h)$
7 for $b \in [B'], \mathbf{y}_{b,[t]}^g \in \mathcal{Y}_t^g$ and $\lambda > 0$
    // perform one step of beam search
8 $Y_{[t]}^g \leftarrow$ $\operatorname{argmax}_{(\mathbf{y}_{1,[t]}^g,\ldots,\mathbf{y}_{B',[t]}^g)} \sum_{b\in[B']} \Theta_t(\mathbf{y}_{b,[t]}^g)$
9 s.t. $\mathbf{y}_{i,[t]}^g \neq \mathbf{y}_{j,[t]}^g \ \forall i \neq j$
10 Return set of B solutions, $Y_{[T]} = \bigcup_{g=1}^G Y_{[T]}^g$

As expected, we find that a higher value of λ produces a more diverse list; however, very large values of λ can overpower model score and result in grammatically incorrect outputs. We set λ via grid search over a range of values to maximize oracle accuracies achieved on the validation set. We find a wide range of λ values (0.2 to 0.8) work well for most tasks and datasets with which we experimented.

Choice of Diversity Function (Δ). We defined $\Delta(y, Y)$ as a dissimilarity function between a sequence y and a set of sequences Y. In Section 4, we illustrated a simple hamming diversity of this form that penalizes selection of tokens proportionally to the number of time it was used in previous groups. However, this factorized diversity term can take various forms, encoding different notions of diversity – with hamming diversity being the simplest.

For language models, we consider various forms like cumulative diversity (time-averaged hamming diversity), n-gram diversity (discourages n-grams occurring in previous groups) and neural embedding based diversity functions that softly compute dissimilarity using average distances in a semantic space (specifically Word2Vec (Mikolov et al. 2013) space). While all diversity functions result in DBS significantly outperforming BS, we empirically find that the default hamming diversity function to be most effective (see Fig. 3) and report results based on this diversity measure.

Beam Size (B). While larger beam sizes often lead to better exploration of the search space, it is computationally expensive. We find that promoting diversity in the decoded lists via DBS leads to a more efficient usage of the beam budget – for instance, to achieve a SPICE score of $\sim$10.89 DBS requires a beam size of 40 compared to the 100 needed by BS.

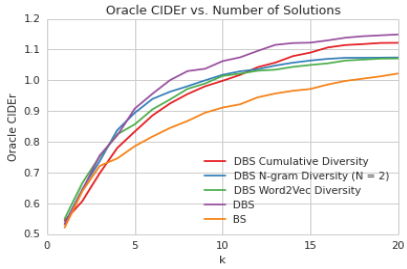

Figure 3: On the PASCAL-50S dataset, we compare the oracle CIDEr@k (Vedantam, Lawrence Zitnick, and Parikh 2015) for lists sampled using a beam size of 20. While all variants of DBS significantly outperform DBS, we find that using simple hamming diversity performs best. We find similar results across other metrics.

5 Experiments

In this section, we evaluate our approach on image captioning, visual question generation and machine translation tasks to demonstrate both its effectiveness against baselines and its general applicability to any inference currently supported by beam search. Further, we explore the role of diversity in generating language from complex images. We first explain the baselines and evaluations used in the following sections.

Baselines. Apart from classical beam search (BS), we compare our method with two related methods;

- Li and Jurafsky (2016) modify BS by introducing an intra-sibling rank. For each partial solution, the set of $|\mathcal{V}|$ beam extensions are sorted and assigned intra-sibling ranks $k \in [|\mathcal{V}|]$ in order of decreasing log probabilities. The log probability of an extension is then reduced in proportion to its rank, and continuations are re-sorted under these modified log probabilities to select the top B 'diverse' beam extensions, and
- Li et al. (2015) train an additional unconditioned target sequence model $U(\mathbf{y})$ and perform BS decoding on an augmented objective $P(\mathbf{y}|x) - \lambda U(\mathbf{y})$, penalizing input-independent decodings.

We compare to our own implementations of these methods as none are publicly available. Both (Li and Jurafsky 2016) and (Li et al. 2015) develop and use re-rankers to pick a single solution from the generated lists. Since we are interested in evaluating the quality and diversity of the entire set of decoded outputs, we simply rank by log-probability.

Hyperparameters. We set all hyperparameters for DBS and the baseline methods by maximizing oracle SPICE via grid-search on a held out validation set for each experiment.

Evaluation Metrics. We evaluate the performance of the generated lists using the following two metrics:

- *Sequence Metrics*: Task-specific metrics that measure the quality of a sentence against ground truth sequences. We use SPICE (Anderson et al. 2016) for image captioning and BLEU (Papineni et al. 2002) for machine translation.
- *Oracle Performance*: Oracle or top k performance w.r.t. some sequence metric is the maximum value of the metric achieved over a list of k potential solutions. Oracle performance is an upper bound on the performance of any re-ranker, measuring the possible impact of diversity.
- *Distinct n-Grams*: We count the number of distinct n-grams present in the list of generated outputs. Similar to (Li et al. 2015), we divide these counts by the total number of words generated to bias against long sentences.

Simultaneous improvements in all metrics indicate that output sequences have increased in diversity without sacrificing fluency or correctness with respect to the target tasks.

5.1 Estimating Image Complexity

One implicit thesis of this work is that language grounded in complex scenes is more diverse. To evaluate this claim, we assess if diversity in language generation leads to larger improvements on more complex images.

One notion of image complexity is studied by Ionescu *et al*. (Ionescu et al. 2016), defining a difficulty score as the human response time for solving a visual search task for images in PASCAL-50S (Vedantam, Lawrence Zitnick, and Parikh 2015). Using the data from (Ionescu et al. 2016), we train a Support Vector Regressor on ResNet (He et al. 2016) features to predict this difficulty score. This model achieves a 0.41 correlation with the ground truth (comparable to the best model of (Ionescu et al. 2016) at 0.47).

To evaluate the relationship between image complexity and performance gains from diverse decoding, we use this trained predictor to estimate a difficulty score s for each image in the COCO (Lin et al. 2014) dataset. We compute the mean ($\mu = 3.3$) and standard deviation ($\sigma = 0.61$) and divide the images into three bins, `Simple` ($s \leq \mu - \sigma$), `Average` ($\mu - \sigma > s < \mu + \sigma$), and `Complex` ($s \geq \mu + \sigma$) consisting of 745, 3416 and 839 images respectively.

Figure 3 shows some sample `Simple`, `Average`, and `Complex` images from the PASCAL-50S dataset. While simple images like close-up pictures of cats may only be described in a handful of ways by human captioners (first column), complex images with multiple objects and interactions will be described in many different ways depending on what is the focus of the captioner (last column).

In the following experiments, we show that improvements from DBS are greater for more complex images.

5.2 Image Captioning

We begin by validating our approach on the COCO (Lin et al. 2014) image captioning task consisting of five human generated captions per image. We use the public splits as in (Karpathy and Fei-Fei 2015) and train a captioning model (Vinyals et al. 2015) using the `neuraltalk2`[1] codebase. We compare decoding methods on this model.

Results by Image Complexity. Each approach produces $B = 20$ candidates that are ranked by log-probability to compute Oracle SPICE@k for different values of k. We note that at $k = 1$ this is directly the standard SPICE evaluation metric. From Table 1, we can see that as the complexity of images increases DBS outperforms standard beam search (difference shown in parentheses) and other baselines

[1] https://github.com/karpathy/neuraltalk2

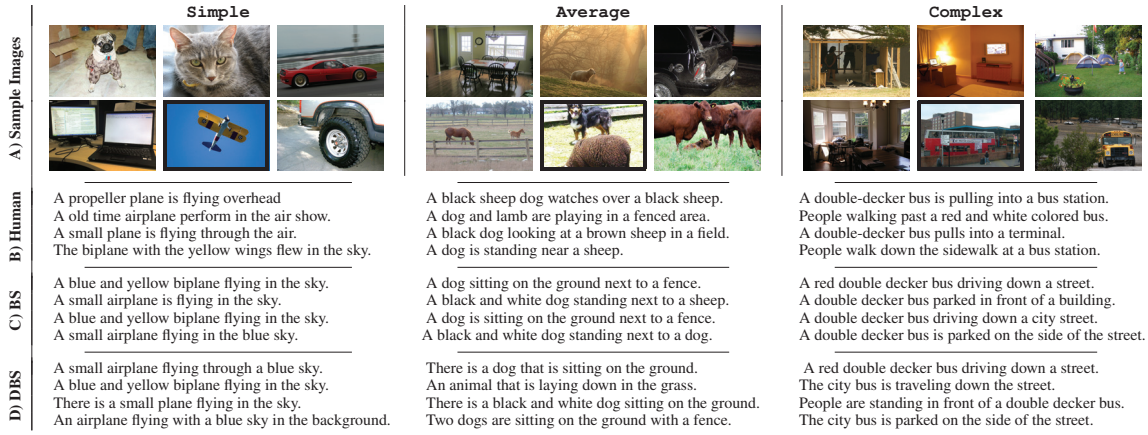

Figure 4: *A)* Sample PASCAL-50S images of different difficulty. Simple images are often close-ups of single objects while complex images involve multiple objects in a wider view. *B)* Random human captions for the black-bordered images. Complex images have more varied captions than simpler images. *C*) which are not captured well by beam search compared to *D)* DBS.

by larger margins for all values of k. For example, at Oracle Spice@20, DBS achieves significant improvements over BS of 0.67, 0.91, and 1.13 for `Simple`, `Average`, and `Complex` images respectively. While DBS improves over BS in all settings, complex images benefit even more from diversity-inducing inference than simple images.

Overall Results. The top half of Table 1 shows results and distinct n-gram statistics on this task. We observe that DBS outperforms BS and Li et al. (2015) while being comparable to or slightly better than Li and Jurafsky (2016) that uses an additional language model. DBS also generates more distinct n-grams than other baselines and produces slightly longer captions (an almost 300% increase in the number of 4-grams and +0.97 words on average w.r.t. BS).

Evaluating Under Greater Human Supervision. While the COCO dataset's size enables powerful captioning models to be trained, with only five captions per image it represents a sparse sample that may miss much of the diversity in visually grounded natural language. So we also evaluate our COCO trained model on the PASCAL-50S (Vedantam, Lawrence Zitnick, and Parikh 2015) dataset which consists of 1000 images with 50 captions each. Having ten times as many captions per image than COCO, the PASCAL-50S dataset captures greater diversity in human annotations and we would expect to see diverse decoding have a greater impact in this setting. We keep 200 random images as a validation set for tuning and evaluate on the remaining images.

Table 2 shows results on this transfer task. As expected, we observe that gains over standard decoding on PASCAL-50S are more pronounced than on COCO (2.74% vs. 6.33% improvement over BS in SPICE@20 using DBS). As in the above experiments, we find that DBS outperforms the baseline methods and produces more diverse captions. Moreover, we note that DBS finds top-1 solutions with higher log-probability on average – obtaining an average maximum log probability of -6.53 opposed to -6.91 found by BS at the same beam width. This empirical evidence suggests that using DBS instead of BS may lead to lower approximate inference error in some cases in addition to improved diversity.

	Method	SPICE	Oracle SPICE@k			Distinct n-Grams			
			@5	@10	@20	n = 1	2	3	4
COCO	BS	16.27	22.96	25.14	27.34	0.40	1.51	3.25	5.67
	Li and Jurafsky (2016)	16.35	22.71	25.23	27.59	0.54	2.40	5.69	8.94
	DBS	**16.783**	**23.08**	**26.08**	**28.09**	**0.56**	**2.96**	**7.38**	**13.44**
	Li et al. (2015)	16.74	23.27	26.10	27.94	0.42	1.37	3.46	6.10

	Method	SPICE	Oracle SPICE@k (Gain over BS)		
			@5	@10	@20
Simple	BS	17.28 (0)	24.32 (0)	26.73 (0)	28.7 (0)
	Li and Jurafsky (2016)	17.12 (-0.16)	24.17 (-0.15)	26.64 (-0.09)	29.28 (0.58)
	DBS	**17.42 (0.14)**	**24.44 (0.12)**	**26.92 (0.19)**	**29.37 (0.67)**
	Li et al. (2015)	17.38 (0.1)	24.48 (0.16)	26.82 (0.09)	29.21 (0.51)
Average	BS	15.95 (0)	22.51 (0)	24.8 (0)	26.55 (0)
	Li and Jurafsky (2016)	16.19 (0.24)	22.59 (0.08)	24.98 (0.18)	27.23 (0.68)
	DBS	**16.28 (0.33)**	**22.65 (0.14)**	**25.08 (0.28)**	**27.46 (0.91)**
	Li et al. (2015)	16.22 (0.27)	22.61 (0.1)	25.01 (0.21)	27.12 (0.57)
Complex	BS	16.39 (0)	22.62 (0)	24.91 (0)	27.23 (0)
	Li and Jurafsky (2016)	16.55 (0.16)	22.55 (-0.07)	25.18 (0.27)	27.57 (0.34)
	DBS	**16.75 (0.36)**	**22.81 (0.19)**	**25.25 (0.34)**	**28.36 (1.13)**
	Li et al. (2015)	16.69 (0.3)	22.69 (0.07)	25.16 (0.25)	27.94 (0.71)

Table 1: *Top:* Oracle SPICE@k and distinct n-grams on the COCO image captioning task at $B = 20$. While we report SPICE, we observe similar trends in other metrics (reported in the supplement). *Bottom:* Breakdown of results by difficulty class, highlighting the relative improvement over BS.

Human Preference by Difficulty. To further establish the effectiveness of our method, we evaluate human preference between captions decoded using DBS and BS. In this forced-choice test, DBS captions were preferred over BS 60% of the time by human annotators. Further, they were preferred about 50%, 69% and 83% of the times for `Simple`, `Average` and `Difficult` images respectively. Furthermore, we observe a positive correlation ($\rho = 0.73$) between difficulty scores and humans preferring DBS to BS. Further details about this experiment are provided in the supplement.

	Method	SPICE	Oracle SPICE@k			Distinct n-Grams			
			@5	@10	@20	n = 1	2	3	4
PASCAL-50S	BS	4.93	7.04	7.94	8.74	0.12	0.57	1.35	2.50
	Li and Jurafsky (2016)	5.08	7.24	8.09	8.91	0.15	0.97	2.43	5.31
	DBS	**5.357**	**7.357**	**8.269**	**9.293**	**0.18**	**1.26**	**3.67**	**7.33**
	Li et al. (2015)	5.12	7.17	8.16	8.56	0.13	1.15	3.58	8.42

Table 2: Oracle SPICE@k and distinct n-grams PASCAL-50S at $B = 20$. While we report SPICE, we observe similar trends in other metrics.

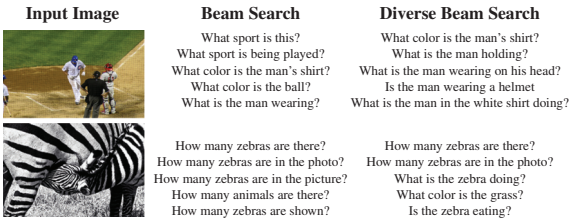

Figure 5: Qualitative results on Visual Question Generation. DBS generates more varied question types than BS.

5.3 Visual Question Generation

We also report results on Visual Question Generation (VQG) (Mostafazadeh et al. 2016), where a model is trained to produce questions *about an image*. Generating visually focused questions requires reasoning about multiple problems that are central to vision – *e.g.*, object attributes, relationships between objects, and natural language. Similar to captioning, there are many sensible questions for a given image.

The VQG dataset (Mostafazadeh et al. 2016) consists of 5 human-generated questions per image for 5000 images from COCO (Lin et al. 2014). We use a model similar to the one used for captioning, except that it is now trained to output questions rather than captions. Similar to previous results, using beam search to sample outputs results in similarly worded question while DBS decoded questions ask about multiple details of the image (see Fig. 5).

We show quantitative evaluations in Table 3 for the VQG dataset as a whole and when partitioned by image difficulty. We find DBS significantly outperforms the baseline methods on this task both on standard metrics (SPICE) and measure of diversity. We also observe that gap between DBS and the baseline methods is more pronounced than in the captioning task and attribute this to the increased variety of possible visually grounded questions compared to captions which often describe only a few major salient objects. The general trend that more complex images benefit more from diverse decoding also persists in this setting.

5.4 Machine Translation

Dataset and Models. We use the English-French parallel data from the *europarl* corpus as the training set. We report results on *news-test-2013* and *news-test-2014* and use the *newstest2012* to tune DBS parameters. We train a encoder-decoder architecture as proposed in (Bahdanau, Cho, and Bengio 2014) using the `dl4mt-tutorial`[2] code repository. The encoder consists of a bi-directional recurrent network (Gated Recurrent Unit) with attention. From Table 4, we see that DBS consistently outperforms standard baselines with respect to both quality and diversity – highlighting the general applicability of DBS to sequence decoding tasks.

	Method	SPICE	Oracle SPICE@k			Distinct n-Grams			
			@5	@10	@20	n = 1	2	3	4
VQG	BS	15.17	21.96	23.16	26.74	0.31	1.36	3.15	5.23
	Li and Jurafsky (2016)	15.45	22.41	25.23	27.59	0.34	2.40	5.69	8.94
	DBS	**16.49**	**23.11**	**25.71**	**27.94**	**0.43**	**2.17**	**6.49**	**12.24**
	Li et al. (2015)	16.34	22.92	25.12	27.19	0.35	1.56	3.69	7.21

	Method	SPICE	Oracle SPICE@k (Gain over BS)		
			@5	@10	@20
Simple	BS	16.04 (0)	21.34 (0)	23.98 (0)	26.62 (0)
	Li and Jurafsky (2016)	16.12 (0.12)	21.65 (0.31)	24.64 (0.66)	26.68 (0.04)
	DBS	**16.42 (0.38)**	**22.44 (1.10)**	**24.71 (0.73)**	**26.73 (0.13)**
	Li et al. (2015)	16.18 (0.14)	22.18 (0.74)	24.16 (0.18)	26.23 (-0.39)
Average	BS	15.29 (0)	21.61 (0)	24.12 (0)	26.55 (0)
	Li and Jurafsky (2016)	16.20 (0.91)	21.90 (0.29)	25.61 (1.49)	27.41 (0.86)
	DBS	**16.63 (1.34)**	**22.81 (1.20)**	**24.68 (0.46)**	**27.10 (0.55)**
	Li et al. (2015)	16.07 (0.78)	22.12 (-0.49)	24.34 (0.22)	26.98 (0.43)
Complex	BS	15.78 (0)	22.41 (0)	24.48 (0)	26.87 (0)
	Li and Jurafsky (2016)	16.82 (1.04)	23.20 (0.79)	25.48 (1.00)	27.12 (0.25)
	DBS	**17.25 (1.47)**	**23.35(1.13)**	**26.19 (1.71)**	**28.01 (1.03)**
	Li et al. (2015)	17.10 (1.32)	23.31 (0.90)	26.01 (1.53)	27.92 (1.05)

Table 3: *Top:* Oracle SPICE@k and distinct n-grams on the VQG task at $B = 20$. *Bottom:* Results by difficulty class, highlighting the relative improvement over BS.

Method	BLEU-4	Oracle BLEU-4@k			Distinct n-Grams			
		@5	@10	@20	n=1	2	3	4
BS	13.52	16.67	17.63	18.44	0.04	0.75	2.10	3.23
Li and Jurafsky (2016)	13.63	17.11	17.50	18.34	0.04	0.81	2.92	4.61
DBS	**13.69**	**17.51**	**17.80**	**18.77**	**0.06**	**0.95**	**3.67**	**5.54**
Li et al. (2015)	13.40	17.54	17.97	18.86	0.04	0.86	2.76	4.31

Table 4: Quantitative results on En-Fr machine translation on the newstest-2013 dataset (at $B = 20$). We find similar trends hold for other BLEU metrics as well.

6 Conclusion

In this work, we propose *Diverse Beam Search* that modifies classical Beam Search decoding with a diversity-augmented sequence decoding objective. Our algorithm is a 'doubly greedy' approximate algorithm that minimizes this augmented objective to produce diverse sequence decodings. DBS consistently outperforms beam search and other baselines across all our experiments without *substantial extra computation* or any *task-specific overhead*. DBS is *task-agnostic* and we demonstrate the effectiveness of our method on two visually grounded language generation tasks – captioning and question generation – as well as on a machine translation task. In the interest of transparent and reproducible research, our implementation in multiple deep learning frameworks will be made publicly available.

[2]https://github.com/nyu-dl/dl4mt-tutorial

References

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2016. Spice: Semantic propositional image caption evaluation. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Batra, D.; Yadollahpour, P.; Guzman-Rivera, A.; and Shakhnarovich, G. 2012. Diverse M-Best Solutions in Markov Random Fields. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Das, A.; Kottur, S.; Gupta, K.; Singh, A.; Yadav, D.; Moura, J. M.; Parikh, D.; and Batra, D. 2017. Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Gimpel, K.; Batra, D.; Dyer, C.; and Shakhnarovich, G. 2013. A systematic exploration of diversity in machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Graves, A.; Mohamed, A.; and Hinton, G. E. 2013. Speech recognition with deep recurrent neural networks. abs/1303.5778.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Ionescu, R. T.; Alexe, B.; Leordeanu, M.; Popescu, M.; Papadopoulos, D.; and Ferrari, V. 2016. How hard can it be? Estimating the difficulty of visual search in an image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kirillov, A.; Savchynskyy, B.; Schlesinger, D.; Vetrov, D.; and Rother, C. 2015. Inferring m-best diverse labelings in a single one. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kirillov, A.; Shekhovtsov, A.; Rother, C.; and Savchynskyy, B. 2016. Joint m-best-diverse labelings as a parametric submodular minimization. In *Advances in Neural Information Processing Systems*, 334–342.

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

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2015. A diversity-promoting objective function for neural conversation models. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT)*.

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

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*.

Mostafazadeh, N.; Misra, I.; Devlin, J.; Mitchell, M.; He, X.; and Vanderwende, L. 2016. Generating natural questions about an image. *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*.

Prasad, A.; Jegelka, S.; and Batra, D. 2014. Submodular meets structured: Finding diverse subsets in exponentially-large structured item sets. In *Advances in Neural Information Processing Systems (NIPS)*.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015. Sequence to sequence-video to text. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4534–4542.

Vinyals, O., and Le, Q. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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