A Simple, Fast Diverse Decoding Algorithm for Neural Generation

Jiwei Li, Will Monroe and Dan Jurafsky
Computer Science Department, Stanford University, Stanford, CA, USA
jiweil,wmonroe4,jurafsky@stanford.edu

Abstract

We propose a simple, fast decoding algorithm that fosters diversity in neural generation. The algorithm modifies the standard beam search algorithm by penalizing hypotheses that are siblings—expansions of the same parent node in the search—thus favoring including hypotheses from diverse parents. We evaluate the model on three neural generation tasks: dialogue response generation, abstractive summarization, and machine translation. We also describe an extended model that uses reinforcement learning to automatically choose the appropriate level of beam diversity for different inputs or tasks. Simple diverse decoding helps across all three tasks, especially those needing reranking or having diverse ground truth outputs; reinforcement learning offers an additional boost. [1]

1 Introduction

Neural generation models (Sutskever et al., 2014; 0; Cho et al., 2014; Kalchbrenner and Blunsom, 2013) are of growing interest for various applications such as machine translation (Sennrich et al., 2015b; Gulcehre et al., 2015), conversational response generation (Vinyals and Le, 2015; Sordoni et al., 2015; Luan et al., 2016), abstractive summarization (Nallapati et al., 2016; Rush et al., 2015; Chopra et al., 2016), and image caption generation (Chen et al., 2015). Such models are trained by learning to predict an output sequence, and then at test time, the model chooses the best sequence given the input, usually using beam search.

One long-recognized issue with beam search is lack of diversity in the beam: candidates often differ only by punctuation or minor morphological variations, with most of the words overlapping (Macherey et al., 2008; Tromble et al., 2008; Kumar and Byrne, 2004). Lack of diversity can hinder sequence generation quality. For tasks like conversational response generation or image caption generation, there is no one correct answer; the decoder thus needs to explore different paths to various sequences to avoid local minima (Vijayakumar et al., 2016).

Lack of diversity causes particular problems in two-stage re-ranking approaches, in which an N-best list or lattice of candidates is generated using beam search and then re-ranked using features too global or expensive to include in the first beam decoding pass [2]. In neural response generation, a re-ranking step helps avoid generating dull or generic responses (Li et al., 2015a; Sordoni et al., 2015; Shao et al., 2016). Lack of diversity in the N-best list significantly decreases the impact of reranking.[3]

In this paper, we propose a simple, fast, diversity-fostering beam search model for neural decoding; the model can be obtained by changing just one line of beam search code in MATLAB. The algorithm uses standard beam search as its backbone but adds an additional term penalizing siblings—expansions of the same parent node in the search— thus favoring choosing hypotheses from diverse parents (as demonstrated in Figure 1).

The proposed model supports batched decoding using GPUs, significantly speeding up the decoding

[1] This paper includes material from the unpublished manuscript "Mutual Information and Diverse Decoding Improve Neural Machine Translation" (Li and Jurafsky, 2016).

[2] E.g., position bias or bilingual attention symmetry in MT (Cohn et al., 2016) or global discourse features in summarization.

[3] Shao et al. (2016), for example, find that re-ranking heuristics in conversational response generation work for shorter responses but not for long responses, since the beam N-best list for long responses are mostly identical even with a large beam.

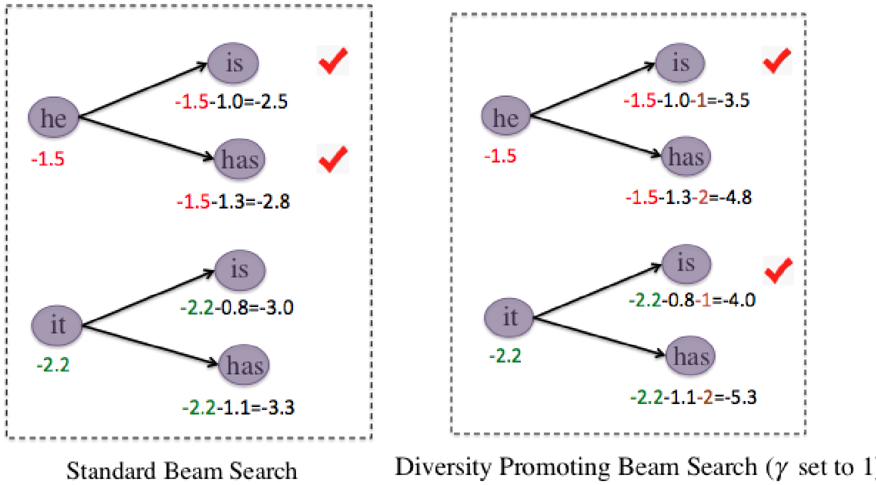

Figure 1: An illustration of standard beam search and the proposed diversity-promoting beam search. γ denotes the hyperparameter for penalizing intra-sibling ranking. Scores are made up for illustration purposes.

process compared to other diversity fostering models for phrase-based MT systems (Macherey et al., 2008; Tromble et al., 2008; Kumar and Byrne, 2004; Devlin and Matsoukas, 2012). To show the generality of the model, we evaluate it on three neural generation tasks—conversational response generation, abstractive summarization and machine translation– demonstrating that the algorithm generates better outputs due to considering more diverse sequences. We discuss which properties of these various tasks make them more or less likely to be helped by diverse decoding. We also propose a more sophisticated variant that uses reinforcement learning to automatically adjust the diversity rate for different inputs, yielding an additional performance boost.

2 Related Work

Diverse decoding has been sufficiently explored in phrase-based MT (Huang, 2008; Finkel et al., 2006), including use of compact representations like lattices and hypergraphs (Macherey et al., 2008; Tromble et al., 2008; Kumar and Byrne, 2004), "traits" like translation length (Devlin and Matsoukas, 2012), bagging/boosting (Xiao et al., 2013), blending multiple systems (Cer et al., 2013), and sampling translations proportional to their probability (Chatterjee and Cancedda, 2010). The most relevant is work from Gimpel et al. (2013) and Batra et al. (2012) that produces diverse N-best lists by adding a dissimilarity function based on N-gram overlaps, distancing the current translation from already-generated ones by choosing translations that have high scores but are distinct from previous ones. While we draw on these intuitions, these existing diversity-promoting algorithms are tailored to phrase-based translation frameworks and not easily transplanted to neural MT decoding, which requires batched computation.

Some recent work has looked at decoding for neural generation. Cho (2016) proposed a meta-algorithm that runs in parallel many chains of the noisy version of an inner decoding algorithm. Vijayakumar et al. (2016) proposed a diversity-augmented objective for image caption generation akin to a neural version of Gimpel et al. (2013). Shao et al. (2016) used a stochastic search algorithm that reranks the hypothesis segment by segment, which injects diversity earlier in the decoding process.

The proposed RL based algorithm is inspired by a variety of recent reinforcement learning approaches in NLP for tasks such as dialogue (Dhingra et al., 2016), word compositions (Yogatama et al., 2016), machine translation (Ranzato et al., 2015), neural model visualization (Lei et al., 2016), and coreference (Clark and Manning, 2016).

3 Diverse Beam Decoding

In this section, we introduce the proposed algorithm. We first go over the vanilla beam search method and then detail the proposed algorithm which fosters diversity during decoding.

3.1 Basics

Let X denote the source input, which is the input dialogue history for conversational response generation or a source sentence for machine translation. The input X is mapped to a vector representation, which is used as the initial input to the decoder. Each X is paired with a target sequence Y, which corresponds to a dialogue utterance in response generation or a target sentence in machine translation. $Y = \{y_1, y_2, ..., y_{n_y}\}$ consists a sequence of n_y words. A neural generation model defines a distribution over outputs and sequentially predicts tokens using a softmax function:

$$p(Y|X) = \prod_{t=1}^{n_y} p(y_t|X, y_1, y_2, ..., y_{t-1})$$

At test time, the goal is to find the sequence Y^* that maximizes the probability given input X:

$$Y^* = \arg\max_X p(Y^*|X) \tag{1}$$

3.2 Standard Beam Search for N-best lists

N-best lists are standardly generated from a model of $p(Y|X)$ using a beam search decoder. As illustrated in Figure 1, at time step $t-1$ in decoding, the decoder keeps track of K hypotheses, where K denotes the beam size, and their scores $S(Y_{t-1}|X) = \log p(y_1, y_2, ..., y_{t-1}|X)$. As it moves on to time step t, it expands each of the K hypotheses (denoted as $Y_{t-1}^k = \{y_1^k, y_2^k, ..., y_{t-1}^k\}$, $k \in [1, K]$) by selecting the top K candidate expansions, each expansion denoted as $y_t^{k,k'}$, $k' \in [1, K]$, leading to the construction of $K \times K$ new hypotheses:

$$[Y_{t-1}^k, y_t^{k,k'}], k \in [1, K], k' \in [1, K]$$

The score for each of the $K \times K$ hypotheses is computed as follows:

$$S(Y_{t-1}^k, y_t^{k,k'}|x) = S(Y_{t-1}^k|x) + \log p(y_t^{k,k'}|x, Y_{t-1}^k) \tag{2}$$

In a standard beam search model, the top K hypotheses are selected (from the $K \times K$ hypotheses computed in the last step) based on the score $S(Y_{t-1}^k, y_t^{k,k'}|x)$. The remaining hypotheses are ignored when the algorithm proceeds to the next time step.

3.3 Generating a Diverse N-best List

Unfortunately, the N-best lists outputted from standard beam search are a poor surrogate for the entire search space (Finkel et al., 2006; Huang, 2008). The beam search algorithm can only keep a small proportion of candidates in the search space, and most of the generated translations in N-best list are similar. Our proposal is to increase diversity by changing the way $S(Y_{t-1}^k, y_t^{k,k'}|x)$ is computed, as shown in Figure 1. For each of the hypotheses Y_{t-1}^k (*he* and *it*), we generate the top K translations $y_t^{k,k'}$, $k' \in [1, K]$ as in the standard beam search model. Next, we rank the K translated tokens generated from the same parental hypothesis based on $p(y_t^{k,k'}|x, Y_{t-1}^k)$ in descending order: *he is* ranks first among *he is* and *he has*, and *he has* ranks second; similarly for *it is* and *it has*.

We then rewrite the score for $[Y_{t-1}^k, y_t^{k,k'}]$ by adding an additional term $\gamma k'$, where k' denotes the ranking of the current hypothesis among its siblings (1 for *he is* and *it is*, 2 for *he has* and *it has*).

$$\hat{S}(Y_{t-1}^k, y_t^{k,k'}|x) = S(Y_{t-1}^k, y_t^{k,k'}|x) - \gamma k' \tag{3}$$

We call γ the *diversity rate*; it indicates the degree of diversity one wants to integrate into the beam search model.

The top K hypotheses are selected based on $\hat{S}(Y_{t-1}^k, y_t^{k,k'}|x)$ as we move on to the next time step. By adding the additional term $\gamma k'$, the model punishes lower-ranked hypotheses among siblings (hypotheses descended from the same parent). When we compare newly generated hypotheses descended from different ancestors, the model gives more credit to top hypotheses from each of the different ancestors. For instance, even though the original score for *it is* is lower than *he has*, the model favors the former as the latter is more severely punished by the intra-sibling ranking part $\gamma k'$. The model thus generally favors choosing hypotheses from diverse parents, leading to a more diverse N-best list. The proposed model is straightforwardly implemented with a minor adjustment to the standard beam search.

4 Automatically Learning Diversity Rate

One disadvantage of the algorithm described above is that a fixed diversity rate γ is applied to all examples. Yet the optimal diversity rate could vary from instance to instance, and too high a diversity could even be detrimental if it pushes the decoding model too far from the beam search scores. Indeed, Shao et al. (2016) find that in response generation, standard beam search works well for short responses but deteriorates as the sequence gets longer, while Vijayakumar et al. (2016) argue in image caption generation that diverse decoding is beneficial for images with many objects, but not images with few objects.

A good diverse decoding algorithm should have the ability to automatically adjust its diversity rates for different inputs—for example, using small diversity rates for images with fewer objects but larger rates for those with more objects. We propose a reinforcement learning-based algorithm called *diverseRL* that is capable of learning different γ values for different inputs.

4.1 Model

We first define a list Γ that contains the values that γ can take. For example, Γ might consist of the 21 values in the range [0,1] at regularly spaced intervals 0.05 apart.[4] Our main idea is to use reinforcement learning (policy gradient methods) to discover the best diversity rate $\gamma(X)$ for a given input X with respect to the final evaluation metric. For each input X, we parameterize the action of choosing an associated diversity rate $\gamma(X)$ by a policy network $\pi(\gamma(X) = \gamma'|X)$ which is a distribution over the $|\Gamma|$ classes.[5] We first map the input X to a vector representation h_X using a recurrent net[6], and then map h_X to a policy distribution over different values of γ using a softmax function:

$$\pi(\gamma(X) = \gamma'|X) = \frac{\exp(h_X^T \cdot h_{\gamma'})}{\sum_{j=1}^{j=|\Gamma|} \exp(h_X^T \cdot h_{\Gamma_j})} \tag{4}$$

Given an action, namely a choice of γ' for $\gamma(X)$, we start decoding using the proposed diverse decoding algorithm and obtain an N-best list. Then we pick the best output—the output with the largest reranking score, or the output with the largest probability if no reranking is needed. Using the selected output, we compute the evaluation score (e.g., BLEU) denoted $R(\gamma(X) = \gamma')$, and this score is used as the reward[7] for the action of choosing diversity rate $\gamma(X) = \gamma'$.

We use the REINFORCE algorithm (Williams, 1992), a kind of policy gradient method, to find the optimal diversity rate policy by maximizing the expectation of the final reward, denoted as follows:

$$E_{\pi(\gamma(X)=\gamma'|X)}[R(\gamma(X) = \gamma'))] \tag{5}$$

The expectation is approximated by sampling from π and the gradient is computed based on the likelihood ratio (Glynn, 1987; VM et al., 1968):

$$\nabla E(\theta) = [R(\gamma(X) - b]\nabla \log \pi(\gamma(X) = \gamma'|X) \tag{6}$$

where b denotes the baseline value.[8] The model is trained to take actions (choosing a diversity rate γ) that will lead to the highest value of final rewards.

4.2 Training and Testing

To learn the policy $\pi(\gamma(X) = \gamma'))$, we take a pre-trained encoder-decoder model and run additional epochs over the training set in which we keep the encoder-decoder parameters fixed and do the diverse beam search using $\gamma(X)$ sampled from the policy distribution. Because decoding is needed for every training sample, training is extremely time-consuming. Luckily, since the sentence composition model involved in $\pi(\gamma(X) = \gamma|X)$ shares parameters with the pre-trained encoder-decoder model, the only parameters needed to be learned are those in the softmax function in Eq. 4, the number of which is relatively small. We therefore only take a small fraction of training examples (around 100,000 instances).

Special attention is needed for tasks in which feature-based reranking is used for picking the final output: feature weights will change as γ changes

[4] This is just for illustration purpose. One can define any set of diversity rate values.

[5] This means the probability of $\gamma(X)$ taking on two similar values (e.g., 0.05 and 0.1) are independent. An alternative is to make γ continuous. However, we find that using discrete values is good enough because of the large amount of training data, and discrete values are easier to implement.

[6] This recurrent net shares parameters with the standard generation model.

[7] This idea is inspired by recent work (Ranzato et al., 2015) that uses BLEU score as reward in reinforcement learning for machine translation. Our focus is different since we are only interested in learning the policy to obtain diversity rates $\gamma(X)$.

[8] The baseline value is estimated using another neural model that takes as input X and outputs a scalar b denoting the estimation of the reward. The baseline model is trained by minimizing the mean squared loss between the estimated reward b and actual cumulative reward r, $||r - b||^2$. We refer the readers to (Ranzato et al., 2015; Zaremba and Sutskever, 2015) for more details. The baseline estimator model is independent from the policy models and the error is not backpropagated back to them.

because those weights are tuned based on the decoded N-best list for dev set, usually using MERT (Och, 2003). Different γ will lead to different dev set N-best lists and consequently different feature weights. We thus adjust feature weights using the dev set after every 10,000 instances. Training takes roughly 1 day.

5 Experiments

We conduct experiments on three different sequence generation tasks: conversational response generation, abstractive summarization and machine translation, the details of which are described below.

5.1 Conversational Response Generation

5.1.1 Dataset and Training

We used the OpenSubtitles (OSDb) dataset (Tiedemann, 2009), an open-domain movie script dataset containing roughly 60M-70M scripted lines spoken by movie characters. Our models are trained to predict the current turn given the preceding ones. We trained a two-layer encoder-decoder model with attention (0; Luong et al., 2015), with 512 units in each layer. We treat the two preceding dialogue utterances as dialogue history, simply concatenating them to form the source input.

To better illustrate in which scenarios the proposed algorithm offers the most help, we construct three different dev-test splits based on reference length, specifically:

- *Natural*: Instances randomly sampled from the dataset.
- *Short*: Instances with target reference length no greater than 6.
- *Long*: Instances with target reference length greater than 16.

Each set contains roughly 2,000 instances.

5.1.2 Decoding and Reranking

We consider the following two settings:

Non-Reranking No reranking is needed. We simply pick the output with highest probability using standard and diverse beam search.

Reranking Following Li et al. (2015a), we first generate an N-best list using vanilla or diverse beam search and rerank the generated responses by combining likelihood $\log p(Y|X)$, backward likelihood $\log p(X|Y)$,[9] sequence length $L(Y)$, and language model likelihood $p(Y)$. The linear combination of $\log(Y|X)$ and $\log p(X|Y)$ is a generalization of the mutual information between the source and the target, which dramatically decreases the rate of dull and generic responses. Feature weights are optimized using MERT (Och, 2003) on N-best lists of response candidates.[10]

	Natural	Short	Long
	Without-Reranking		
vanilla	1.30	1.59	0.89
diverse	1.42	1.63	1.06
	(+9.1%)	(+2.5%)	(+19%)
diverseRL	1.49	1.67	1.03
	(+15%)	(+5.0%)	(+16%)
	With-Reranking		
vanilla	1.88	2.52	1.17
diverse	2.21	2.75	1.68
	(+17%)	(+9.2%)	(+44%)
diverseRL	2.32	2.79	1.70
	(+23%)	(+11%)	(+47%)

Table 1: Response generation: BLEU scores from the vanilla beam search model, the proposed diversity-promoting beam search model and the diverse Beam search + Reinforcement Learning (diverseRL) model on various datasets.

5.1.3 Evaluation

For automatic evaluations, we report (1) BLEU (Papineni et al., 2002), which has widely been used in response generation evaluation; and (2) diversity, which is the number of distinct unigrams and bigrams in generated responses scaled by the total number of generated tokens. This scaling is to avoid favoring long sentences, as described in Li et al. (2016a).

We also do human evaluation, as suggested by Liu et al. (2016). We employ crowdsourced judges to provide evaluations for a random sample of 200 items. Each output pair was ranked by 3 judges, who were asked to decide which of the two outputs was better. They were instructed to prefer outputs that were more specific (relevant) to the context. Ties were permitted. Identical strings were automatically

[9] $\log p(Y|X)$ is trained in a similar way as standard SEQ2SEQ models with only sources and targets being swapped.

[10] We set the minimum length and maximum length of a decoded target to 0.75 and 1.5 times the length of sources. Beam size K is set to 10. We then rerank the N-best list and pick the output target with largest final ranking score.

	Natural	Short	Long
	Without-Reranking		
Vanilla	0.032	0.079	0.009
Diverse	0.049	0.101	0.049
	(+53.2%)	(+27.8%)	(+445%)
	With-Reranking		
Vanilla	0.068	0.252	0.025
Diverse	0.095	0.282	0.070
	(+39.7%)	(+12.2%)	(+180%)

Table 2: Response generation: Diversity scores from standard beam search model and the proposed diversity-promoting beam search model on various datasets.

Diverse-Win	Diverse-Lose	Tie
62%	16%	22%

Table 3: Response generation: The gains over the standard beam search model from diverseRL based on pairwise human judgments.

assigned the same score.

5.1.4 Results

Results for BLEU scores and diversity scores are presented in Tables 1 and 2. Diverse decoding boosts performance across all settings, but more so in the *reranking* settings than the *non-reranking* ones. For *reranking*, the improvement is much larger for longer responses than shorter ones. We suspect that this is because short responses have a smaller search space, and so the vanilla beam search has a sufficiently diverse beam. For longer responses, the generator may get trapped in a local decoding minimum, and as (Shao et al., 2016) pointed out can generate incoherent or even contradictory responses like *"I like fish but I don't like fish"* . Table 4 shows that for these longer responses, these hypotheses generated by standard beam search are nearly identical, dramatically decreasing the impact of the reranking process. [11]

On the *Natural* set, which has the same distribution of length as the full data, we see a performance boost from the *diverseRL* model's ability to dynamically adjust diversity. For the *short* and *long* sets, the improvement from the diverseRL model is less significant, presumably because the dataset has been pre-processed in such a way that examples are more similar to each other in terms of length.

In terms of human evaluation, we find that the *diverseRL* model produces responses with better general quality, winning 62 percent of the time when compared to standard beam search.

[11] Sampling could be another way of generating diverse responses. However, responses from sampling are usually incoherent, especially for long sequences. This is because the error accumulates as decoding goes on. A similar phenomenon has also been observed by Shao et al. (2016).

Input	**As the joint chief of the eight clans, you should do something.**
N-best List (Diverse Beam)	If you want to fight, you must go to the temple. If you want to fight , you can do it yourself . No, i'm not going to do anything. I'm not going to let you go, i'm not going to let you do that. But i don 't want to be a monk
N-best List (Vanilla Beam)	If you want to fight, you must go to the temple . . . . if you want to fight , you must go to the temple to fight . If you want to fight , you must go to the temple . . . If you want to fight , you must go to the police station . If you want to fight , you must go to the palace .
Input	**Well , why can 't you just believe us?**
N-best List (Diverse Beam)	I mean, it 's not like we 're going to be together forever . I mean, you 're the one who 's been stealing my stuff . Because i 'm not a good person . Why can 't we just go to the police and get the hell out of here ? This is a very important moment for us .
N-best List (Vanilla Beam)	I mean , it 's not like we 're going to be together forever . I mean , it 's not like we 're gonna be together forever . I mean , it 's not like we 're going to be together any more . I mean , it 's not like we 're going to be together any more , is it ? I mean , it 's not like we 're going to be friends forever .
Input	**Senator , why didn 't you come to me with this information ?**
N-best List (Diverse Beam)	I didn 't want you to think i was crazy . We could have been killed . Because you 're not a threat to me . You know , i was thinking about the way you handled the situation . If you 're not gonna tell me, i'm not gonna tell you .
N-best List (Vanilla Beam)	I didn 't want you to think i was crazy . I didn 't want to upset you . I didn 't want you to worry . I didn 't want you to find out. I didn 't want you to worry about me .

Table 4: Response generation: Sample responses using the diversity-promoting beam search and vanilla beam search.

5.2 Abstractive Summarization

5.2.1 Training and Dataset

We consider two settings for abstractive summarization. For the first setting (denoted *single*), we follow the protocols described in Rush et al. (2015), in which the source input is the first sentence of the document to summarize. We train a word-level attention model for this setting. Our training dataset consists of 800K pairs.

A good summarization system should have the ability to summarize a large chunk of text, separating wheat from chaff. We thus consider another setting in which each input consists of multiple sentences (denoted *multi*).[12] We train a hierarchical model with

[12] We consider 10 sentences at most for each input. Sentence

	Single	Multi
Without-Reranking		
vanilla	11.2	8.1
diverse	12.1	9.0
With-Reranking		
vanilla	12.4	9.3
diverse	14.0	11.5

Table 5: ROUGE-2 scores from vanilla beam search model and the proposed diversity-promoting beam search model for abstractive summarization.

attention at sentence level (Li et al., 2015b) for generation, which has been shown to yield better results than word-level encoder-decoder models for multi-sentence summarization (Nallapati et al., 2016).

For reranking, we employ various global features taken from or inspired by prior non-neural work in summarization (Daumé III and Marcu, 2006; Nenkova and Vanderwende, 2005; McKeown et al., 1999). Features we consider include (1) average tf-idf score of constituent words in the generated output; (2) *KLSum* (Haghighi and Vanderwende, 2009), which reflects the topic distribution overlap between the entire input document and the generated output;[13] and (3) backward probability $p(X|Y)$, i.e., the probability of generating the entire document given the summary. For evaluation, we report ROUGE-2 scores (Lin, 2004) in Table 5.

The proposed diverse reranking algorithm helps in both cases, but more significantly in the *reranking* setting than the *non-reranking* one. This is because the mechanisms by which diverse decoding helps in the two settings are different: for the *non-reranking* setting, if the conditional entropy in the target distribution is low enough, the decoded string from standard beam search will be close to the global optimum, and thus there won't be much space for improvement. For the *reranking* setting, however, the story is different: it is not just about finding the global optimal output based on the target distribution, but also about incorporating different criteria to make up for facets that are missed by the encoder-decoder model. This requires the N-best list to be diverse for the reranking models to make a significant difference.

We also find an improvement from the *reranking* model using document-level features over the *non-reranking* model. We did not observe a big performance boost from the *diverseRL* model over the standard diverse model (around 0.3 ROUGE score boost) on this task. This is probably because the *diverseRL* model helps most when inputs are different and thus need different diverse decoding rates. In this task, the input documents are all news articles and share similar properties, so a unified diversity rate tuned on the dev set might already be good enough. Results for *diverseRL* are thus omitted for brevity.

Interestingly, we find that the result for the *multi* setting is significantly worse than the *single* setting, which is also observed by Nallapati et al. (2016): adding more sentences leads to worse results. This illustrates the incapability of neural generation models to date to summarize long documents. The proposed diverse decoding model produces a huge performance boost in the *multi* setting.

5.3 Machine Translation

5.3.1 Dataset and Training

The models are trained on the WMT'14 training dataset containing 4.5 million pairs for English-German. We limit our vocabulary to the top 50K most frequent words for both languages. Words not in the vocabulary are replaced by a universal unknown token. We use newstest2013 (3000 sentence pairs) as the development set and report translation performances in BLEU (Papineni et al., 2002) on newstest2014 (2737 sentences). We trained neural SEQ2SEQ models (Sutskever et al., 2014) with attention (Luong et al., 2015; Cho et al., 2014). Unknown words are replaced using methods similar to those of Luong et al. (2015).

5.3.2 Decoding and Reranking

Again we consider both *reranking* settings and *non-reranking* settings, where for *non-reranking* we select the best output using standard beam search. For *reranking*, we first generate a large N-best list using the beam search algorithm.[14] We then rerank

position treated as a word feature which is associated with an embedding to be learned as suggested in Nallapati et al. (2016)

[13] Specifically, the KL divergence between the topic distributions assigned by a variant of the LDA model (Blei et al., 2003) that identifies general, document-specific and topic-specific word clusters.

[14] We use beam size $K = 50$ both for standard beam search and diverse beam search. At each time step of decoding, we are presented with $K \times K$ word candidates. We first add all hypotheses with an *EOS* token generated at the current time step

Non-Reranking	
Vanilla beam	19.8
Diverse	19.8 (+0.04)
Diverse+RL	20.2 (+0.25)
Reranking	
Vanilla	21.5
Diverse	22.1 (+0.6)
Diverse+RL	22.4 (+0.9)

Table 6: BLEU scores from vanilla beam search model and the proposed diversity-promoting beam search model on WMT'14 English-German translation.

the hypotheses using features that have been shown to be useful in neural machine translation (Sennrich et al., 2015a; Gulcehre et al., 2015; Cohn et al., 2016; Cheng et al., 2015), including (1) backward probability $p(X|Y)$; (2) language model probability $p(Y)$ trained from monolingual data (Gulcehre et al., 2015; Gulcehre et al., 2015);[15] (3) bilingual symmetry: the agreement between attentions in German-English and English-German; and (4) target length. Again feature weights are optimized using MERT (Och, 2003) on N-best lists of response candidates in the dev set.

5.3.3 Results

Experimental results are shown in Table 6. First, no significant improvement is observed for the proposed diverse beam search model in the *non-reranking* setting. The explanation is similar to the abstractive summarization task: the vanilla beam search algorithm with large beam size is already good enough at finding the global optimum, and the small benefit from diverse decoding for some examples might even be canceled out by others where diversity rate value is too large. The *diverseRL* model addresses the second issue, leading to a performance boost of +0.25. For the *reranking* setting, the performance boost is more significant, +0.6 for diverse beam search and +0.9 for *DiverseRL*.

to the N-best list. Next, we preserve the top K unfinished hypotheses and move to the next time step. We therefore maintain constant batch size as hypotheses are completed and removed, by adding in more unfinished hypotheses. This allows the size of final N-best list for each input to be much larger than the beam size.

[15] $p(t)$ is trained using a single-layer LSTM recurrent models using monolingual data. We use News Crawl corpora from WMT13 (`http://www.statmt.org/wmt13/translation-task.html`) as additional training data to train monolingual language models. We used a subset of the original dataset which roughly contains 60 millions sentences.

Overall, diverse decoding doesn't seem to improve machine translation as much as it does summarization and response generation. We suspect this is due to the very low entropy of the target distribution (perplexity less than 6), so standard beam search is already fairly strong.[16]

6 Discussion

In this paper, we introduce a general diversity-promoting decoding algorithm for neural generation. The model adds an intra-sibling ranking term to the standard beam search algorithm, favoring choosing hypotheses from diverse parents. The proposed model is a general, simple and fast algorithm that will bring a performance boost to all neural generation tasks for which a diverse N-best list is needed.

On top of this approach, we build a more sophisticated algorithm that is capable of automatically adjusting diversity rates for different inputs using reinforcement learning. We find that, at the expense of model complexity and training time (compared with the basic RL-free diverse decoding algorithm), the model is able to adjust its diversity rate to better values, yielding generation quality better than either standard beam search or the basic diverse decoding approach.

Diverse decoding doesn't help all tasks equally, contributing the most in two kinds of tasks: (1) Those with a very diverse space of ground truth outputs (e.g., tasks like response generation), rather than those in which the conditional entropy of the target distribution is already low enough (e.g., machine translation) (2) Tasks for which reranking is needed to incorporate features not included in the first decoder pass, such as document-level abstractive summarization.

Acknowledgements

We would especially like to thank Thang Luong Minh for insightful discussions and releasing relevant code, as well as Sida Wang, Ziang Xie, Chris Manning and other members of the Stanford NLP group for helpful comments and suggestions. We also thank Michel Galley, Bill Dolan, Chris Brockett, Jianfeng Gao and other members of the NLP group at Microsoft Research for helpful discussions. Jiwei Li is supported by a Facebook Fellowship, which

[16] The difference between beam size 2 and beam size 12 in decoding for French-English translation is only around 0.3 BLEU score (Sutskever et al., 2014), which also confirms that the standard decoding algorithm is sufficiently powerful for MT.

we gratefully acknowledge. This work is partially supported by the DARPA Communicating with Computers (CwC) program under ARO prime contract no. W911NF-15-1-0462 and the NSF via IIS-1514268. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the NSF, or Facebook.

References

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

Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. 2012. Diverse m-best solutions in Markov random fields. In *Computer Vision–ECCV 2012*, pages 1–16. Springer.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Daniel Cer, Christopher D Manning, and Daniel Jurafsky. 2013. Positive diversity tuning for machine translation system combination. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 320–328.

Samidh Chatterjee and Nicola Cancedda. 2010. Minimum error rate training by sampling the translation lattice. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 606–615. Association for Computational Linguistics.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Yong Cheng, Shiqi Shen, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2015. Agreement-based joint training for bidirectional attention-based neural machine translation. *arXiv preprint arXiv:1512.04650*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Kyunghyun Cho. 2016. Noisy parallel approximate decoding for conditional recurrent language model. *arXiv preprint arXiv:1605.03835*.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks.

Kevin Clark and Christopher D Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. *arXiv preprint arXiv:1601.01085*.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312. Association for Computational Linguistics.

Jacob Devlin and Spyros Matsoukas. 2012. Trait-based hypothesis selection for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 528–532. Association for Computational Linguistics.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2016. End-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777*.

Jenny Rose Finkel, Christopher D Manning, and Andrew Y Ng. 2006. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 618–626. Association for Computational Linguistics.

Kevin Gimpel, Dhruv Batra, Chris Dyer, Gregory Shakhnarovich, and Virginia Tech. 2013. A systematic exploration of diversity in machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, October*.

Peter W Glynn. 1987. Likelilood ratio gradient estimation: an overview. In *Proceedings of the 19th conference on Winter simulation*, pages 366–375. ACM.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *ACL*, pages 586–594.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, pages 1700–1709.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. Technical report, DTIC Document.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

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

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015b. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Yi Luan, Yangfeng Ji, and Mari Ostendorf. 2016. Lstm based conversation models. *arXiv preprint arXiv:1603.09457*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Wolfgang Macherey, Franz Josef Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 725–734. Association for Computational Linguistics.

Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *AAAI/IAAI*, pages 453–460.

Ramesh Nallapati, Bowen Zhou, Ça glar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

MarcAurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2016. Generating long and diverse responses with neural conversational models.

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

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jörg Tiedemann. 2009. News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

Roy W Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 620–629. Association for Computational Linguistics.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

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

ALEKSAND. VM, VI Sysoyev, and SHEMENEV. VV. 1968. Stochastic optimization. *Engineering Cybernetics*, (5):11.

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

Tong Xiao, Jingbo Zhu, and Tongran Liu. 2013. Bagging and boosting statistical machine translation systems. *Artificial Intelligence*, 195:496–527.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2016. Learning to compose words into sentences with reinforcement learning.

Wojciech Zaremba and Ilya Sutskever. 2015. Reinforcement learning neural turing machines. *arXiv preprint arXiv:1505.00521*, 362.