Learning to Write with Cooperative Discriminators

Ari Holtzman[†] **Jan Buys**[†] **Maxwell Forbes**[†]
Antoine Bosselut[†] **David Golub**[†] **Yejin Choi**[†‡]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[‡]Allen Institute for Artificial Intelligence
{ahai,jbuys,mbforbes,antoineb,golubd,yejin}@cs.washington.edu

Abstract

Despite their local fluency, long-form text generated from RNNs is often generic, repetitive, and even self-contradictory. We propose a unified learning framework that collectively addresses all the above issues by composing a committee of discriminators that can guide a base RNN generator towards more globally coherent generations. More concretely, discriminators each specialize in a different principle of communication, such as Grice's maxims, and are collectively combined with the base RNN generator through a composite decoding objective. Human evaluation demonstrates that text generated by our model is preferred over that of baselines by a large margin, significantly enhancing the overall coherence, style, and information of the generations.

1 Introduction

Language models based on Recurrent Neural Networks (RNNs) have brought substantial advancements across a wide range of language tasks (Jozefowicz et al., 2016; Bahdanau et al., 2015; Chopra et al., 2016). However, when used for long-form text generation, RNNs often lead to degenerate text that is repetitive, self-contradictory, and overly generic, as shown in Figure 1.

We propose a unified learning framework that can address several challenges of long-form text generation by composing a committee of discriminators each specializing in a different principle of communication. Starting with an RNN language model, our framework learns to construct a more powerful generator by training a number of discriminative models that can collectively address limitations of the base RNN generator, and then learns how to weigh these discriminators to form the final decoding objective. These "cooperative" discriminators complement each other and the base language model to form a stronger, more global decoding objective.

Context: The two guards thudded into the door on the other side and began pounding on it, shouting furiously. Scious raised the large bunch of keys then placed them in a large pocket in his coat. "Come, we have to go," he whispered and moved up the corridor, the guards still hammering on the door. Jared looked around. The table next to the room they had been in was bare.

LM: There was a small table set into the center of the table, and a table with two chairs and a table. A chair sat next to the couch. A large man stood in the doorway of the room, his arms crossed over his chest. "What 's up?" He asked.

L2W: Empty. It looked like some kind of *prison*. There were no windows, no doors, no windows. *No sign of anyone in the house.* No one had ever been here before.

Reference: Only a flagon of wine and a couple of mugs plus a crude dice game. Light flickered in the stone corridor from lanterns on the walls. The place was dank and gloomy, moss in scattered patches on the floor and walls. The corridor ended just beyond the door to their former *prison*. *No one else was about.*

Figure 1: Sample generations from an RNN language model (LM) and our system (L2W) conditioning on the context shown on the top. The red, underlined text highlights repetitions, while the *blue, italicized* text highlights details that have a direct semantic parallel in the reference text.

The design of our discriminators are inspired by Grice's maxims (Grice et al., 1975) of quantity, quality, relation, and manner. The discriminators learn to encode these qualities through the selection of training data (e.g. distinguishing a true continuation from a randomly sampled one as in §3.2 Relevance Model), which includes generations from partial models (e.g. distinguishing a true continuation from one generated by a language model as in §3.2 Style Model). The system

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 1638–1649
Melbourne, Australia, July 15 - 20, 2018.

then learns to balance these discriminators by initially weighing them uniformly, then continually updating its weights by comparing the scores the system gives to its own generated continuations and to the reference continuation.

Empirical results (§5) demonstrate that our learning framework is highly effective in converting a generic RNN language model into a substantially stronger generator. Human evaluation confirms that language generated by our model is preferred over that of competitive baselines by a large margin in two distinct domains, and significantly enhances the overall coherence, style, and information content of the generated text. Automatic evaluation shows that our system is both less repetitive and more diverse than baselines.

2 Background

RNN language models learn the conditional probability $P(x_t|x_1, ..., x_{t-1})$ of generating the next word x_t given all previous words. This conditional probability learned by RNNs often assigns higher probability to repetitive, overly generic sentences, as shown in Figure 1 and also in Table 3. Even gated RNNs such as LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Cho et al., 2014) have difficulties in properly incorporating long-term context due to explaining-away effects (Yu et al., 2017b), diminishing gradients (Pascanu et al., 2013), and lack of inductive bias for the network to learn discourse structure or global coherence beyond local patterns.

Several methods in the literature attempt to address these issues. Overly simple and generic generation can be improved by length-normalizing the sentence probability (Wu et al., 2016), future cost estimation (Schmaltz et al., 2016), or a diversity-boosting objective function (Shao et al., 2017; Vijayakumar et al., 2016). Repetition can be reduced by prohibiting recurrence of the trigrams as a hard rule (Paulus et al., 2018). However, such hard constraints do not stop RNNs from repeating through paraphrasing while preventing occasional intentional repetition.

We propose a unified framework to address all these related challenges of long-form text generation by learning to construct a better decoding objective, generalizing over various existing modifications to the decoding objective.

3 The Learning Framework

We propose a general learning framework for conditional language generation of a sequence $\mathbf{y}$ given a fixed context $\mathbf{x}$. The decoding objective for generation takes the general form

$$f_\lambda(\mathbf{x}, \mathbf{y}) = \log(P_{\text{lm}}(\mathbf{y}|\mathbf{x})) + \sum_k \lambda_k s_k(\mathbf{x}, \mathbf{y}), \quad (1)$$

where every s_k is a scoring function. The proposed objective combines the RNN language model probability P_{lm} (§3.1) with a set of additional scores $s_k(\mathbf{x}, \mathbf{y})$ produced by discriminatively trained communication models (§3.2), which are weighted with learned mixture coefficients λ_k (§3.3). When the scores s_k are log probabilities, this corresponds to a Product of Experts (PoE) model (Hinton, 2002).

Generation is performed using beam search (§3.4), scoring incomplete candidate generations $\mathbf{y}_{1:i}$ at each time step i. The RNN language model decomposes into per-word probabilities via the chain rule. However, in order to allow for more expressivity over long range context we do not require the discriminative model scores to factorize over the elements of $\mathbf{y}$, addressing a key limitation of RNNs. More specifically, we use an estimated score $s'_k(\mathbf{x}, \mathbf{y}_{1:i})$ that can be computed for any prefix of $\mathbf{y} = \mathbf{y}_{1:n}$ to approximate the objective during beam search, such that $s'_k(\mathbf{x}, \mathbf{y}_{1:n}) = s_k(\mathbf{x}, \mathbf{y})$. To ensure that the training method matches this approximation as closely as possible, scorers are trained to discriminate prefixes of the same length (chosen from a predetermined set of prefix lengths), rather than complete continuations, except for the entailment module as described in §3.2 Entailment Model. The prefix scores are re-estimated at each time-step, rather than accumulated over beam search.

3.1 Base Language Model

The RNN language model treats the context $\mathbf{x}$ and the continuation $\mathbf{y}$ as a single sequence $\mathbf{s}$:

$$\log P_{\text{lm}}(\mathbf{s}) = \sum_i \log P_{\text{lm}}(\mathbf{s}_i|\mathbf{s}_{1:i-1}). \quad (2)$$

3.2 Cooperative Communication Models

We introduce a set of discriminators, each of which encodes an aspect of proper writing that RNNs usually fail to capture. Each model is trained to discriminate between good and bad generations; we vary the model parameterization and

training examples to guide each model to focus on a different aspect of Grice's Maxims. The discriminator scores are interpreted as classification probabilities (scaled with the logistic function where necessary) and interpolated in the objective function as log probabilities.

Let $D = \{(\mathbf{x_1}, \mathbf{y_1}), \ldots (\mathbf{x_n}, \mathbf{y_n})\}$ be the set of training examples for conditional generation. $D_\mathbf{x}$ denote all contexts and $D_\mathbf{y}$ all continuations. The scoring functions are trained on prefixes of $\mathbf{y}$ to simulate their application to partial continuations at inference time.

In all models the first layer embeds each word w into a 300-dimensional vector $e(w)$ initialized with GloVe (Pennington et al., 2014) pretrained-embeddings.

Repetition Model

This model addresses the maxim of Quantity by biasing the generator to avoid repetitions. The goal of the repetition discriminator is to learn to distinguish between RNN-generated and gold continuations by exploiting our empirical observation that repetitions are more common in completions generated by RNN language models. However, we do not want to completely eliminate repetition, as words do recur in English.

In order to model natural levels of repetition, a score d_i is computed for each position in the continuation $\mathbf{y}$ based on pairwise cosine similarity between word embeddings within a fixed window of the previous k words, where

$$d_i = \max_{j=i-k \ldots i-1} (\text{CosSim}(e(y_j), e(y_i))), \quad (3)$$

such that $d_i = 1$ if y_i is repeated in the window.

The score of the continuation is then defined as

$$s_{\text{rep}}(\mathbf{y}) = \sigma(\mathbf{w}_r^\top \text{RNN}_{\text{rep}}(\mathbf{d})), \quad (4)$$

where $\text{RNN}_{\text{rep}}(\mathbf{d})$ is the final state of a unidirectional RNN ran over the similarity scores $\mathbf{d} = d_1 \ldots d_n$ and $\mathbf{w}_r$ is a learned vector. The model is trained to maximize the ranking log likelihood

$$L_{\text{rep}} = \sum_{\substack{(\mathbf{x},\mathbf{y}_g) \in D, \\ \mathbf{y}_s \sim \text{LM}(\mathbf{x})}} \log \sigma(s_{\text{rep}}(\mathbf{y}_g) - s_{\text{rep}}(\mathbf{y}_s)), \quad (5)$$

which corresponds to the probability of the gold ending $\mathbf{y}_g$ receiving a higher score than the ending sampled from the RNN language model.

Entailment Model

Judging textual quality can be related to the natural language inference (NLI) task of recognizing textual entailment (Dagan et al., 2006; Bowman et al., 2015): we would like to guide the generator to neither contradict its own past generation (the maxim of Quality) nor state something that readily follows from the context (the maxim of Quantity). The latter case is driven by the RNNs habit of paraphrasing itself during generation.

We train a classifier that takes two sentences a and b as input and predicts the relation between them as either *contradiction*, *entailment* or *neutral*. We use the *neutral* class probability of the sentence pair as discriminator score, in order to discourage both contradiction and entailment. As entailment classifier we use the decomposable attention model (Parikh et al., 2016), a competitive, parameter-efficient model for entailment classification.[1] The classifier is trained on two large entailment datasets, SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2017), which together have more than 940,000 training examples. We train separate models based on the vocabularies of each of the datasets we use for evaluation.

In contrast to our other communication models, this classifier cannot be applied directly to the full context and continuation sequences it is scoring. Instead every completed sentence in the continuation should be scored against all preceding sentences in both the context and continuation.

Let $t(\mathbf{a}, \mathbf{b})$ be the log probability of the neutral class. Let $S(\mathbf{y})$ be the set of complete sentences in $\mathbf{y}$, $S_{\text{last}}(\mathbf{y})$ the last complete sentence, and $S_{\text{init}}(\mathbf{y})$ the sentences before the last complete sentence. We compute the entailment score of $S_{\text{last}}(\mathbf{y})$ against all preceding sentences in $\mathbf{x}$ and $\mathbf{y}$, and use the score of the sentence-pair for which we have the least confidence in a *neutral* classification:

$$s_{\text{entail}}(\mathbf{x}, \mathbf{y}) = min_{\mathbf{a} \in S(\mathbf{x}) \cup S_{\text{init}}(\mathbf{y})} t(\mathbf{a}, S_{\text{last}}(\mathbf{y})). \quad (6)$$

Intuitively, we only use complete sentences because the ending of a sentence can easily flip entailment. As a result, we carry over entailment score of the last complete sentence in a generation until the end of the next sentence, in order to maintain the presence of the entailment score in the objective. Note that we check that the current

[1] We use the version without intra-sentence attention.

```
Data: context x, beam size k, sampling temperature t
Result: best continuation
best = None
beam = [x]
for step = 0; step < max_steps; step = step +1 do
    next_beam = []
    for candidate in beam do
        next_beam.extend(next_k(candidate))
        if termination_score(candidate) > best.score
          then
            best = candidate.append(term)
        end
    end
    for candidate in next_beam do
                                        ▷ score with models
        candidate.score += f_λ(candidate)
    end
                              ▷ sample k candidates by score
    beam = sample(next_beam, k, t)
end
if learning then
    update λ with gradient descent by comparing best
      against the gold.
end
return best
```

Algorithm 1: Inference/Learning in the Learning to Write Framework.

sentence is not directly entailed or contradicted by a previous sentence and not the reverse. [2] In contrast to our other models, the score this model returns only corresponds to a subsequence of the given continuation, as the score is not accumulated across sentences during beam search. Instead the decoder is guided locally to continue complete sentences that are not entailed or contradicted by the previous text.

Relevance Model

The relevance model encodes the maxim of Relation by predicting whether the content of a candidate continuation is relevant to the given context. We train the model to distinguish between true continuations and random continuations sampled from other (human-written) endings in the corpus, conditioned on the given context.

First both the context and continuation sequences are passed through a convolutional layer, followed by maxpooling to obtain vector representations of the sequences:

$$a = \text{maxpool}(\text{conv}_a(e(\mathbf{x}))), \quad (7)$$

$$b = \text{maxpool}(\text{conv}_b(e(\mathbf{y}))). \quad (8)$$

[2] If the current sentence entails a previous one it may simply be adding more specific information, for instance: "He hated broccoli. Every time he ate broccoli he was reminded that it was the thing he hated most."

The goal of maxpooling is to obtain a vector representing the most important semantic information in each dimension.

The scoring function is then defined as

$$s_{\text{rel}} = \mathbf{w}_l^T \cdot (a \circ b), \quad (9)$$

where element-wise multiplication of the context and continuation vectors will amplify similarities.

We optimize the ranking log likelihood

$$L_{\text{rel}} = \sum_{\substack{(\mathbf{x},\mathbf{y}_g)\in D,\\ \mathbf{y}_r \sim D_{\mathbf{y}}}} \log \sigma(s_{\text{rel}}(\mathbf{x},\mathbf{y}_g) - s_{\text{rel}}(\mathbf{x},\mathbf{y}_r)), \quad (10)$$

where $\mathbf{y}_g$ is the gold ending and $\mathbf{y}_r$ is a randomly sampled ending.

Lexical Style Model

In practice RNNs generate text that exhibit much less lexical diversity than their training data. To counter this effect we introduce a simple discriminator based on observed lexical distributions which captures writing style as expressed through word choice. This classifier therefore encodes aspects of the maxim of Manner.

The scoring function is defined as

$$s_{\text{bow}}(\mathbf{y}) = \mathbf{w}_s^T \text{maxpool}(e(\mathbf{y})). \quad (11)$$

The model is trained with a ranking loss using negative examples sampled from the language model, similar to Equation 5.

3.3 Mixture Weight Learning

Once all the communication models have been trained, we learn the combined decoding objective. In particular we learn the weight coefficients λ_k in equation 1 to linearly combine the scoring functions, using a discriminative loss

$$L_{\text{mix}} = \sum_{(\mathbf{x},\mathbf{y})\in D} (f_\lambda(\mathbf{x},\mathbf{y}) - f_\lambda(\mathbf{x},\mathcal{A}(\mathbf{x}))^2, \quad (12)$$

where $\mathcal{A}$ is the inference algorithm for beam search decoding. The weight coefficients are thus optimized to minimize the difference between the scores assigned to the gold continuation and the continuation predicted by the current model.

Mixture weights are learned online: Each successive generation is performed based on the current values of λ, and a step of gradient descent is then performed based on the prediction. This has the effect that the objective function changes

	BookCorpus					TripAdvisor				
Model	BLEU	Meteor	Length	Vocab	Trigrams	BLEU	Meteor	Length	Vocab %	Trigrams
L2W	**0.52**	**6.8**	**43.6**	**73.8**	98.9	1.7	11.0	83.8	**64.1**	**96.2**
ADAPTIVELM	**0.52**	6.3	43.5	59.0	92.7	**1.94**	**11.2**	**94.1**	52.6	92.5
CACHELM	0.33	4.6	37.9	31.0	44.9	1.36	7.2	52.1	39.2	57.0
SEQ2SEQ	0.32	4.0	36.7	23.0	33.7	1.84	8.0	59.2	33.9	57.0
SEQGAN	0.18	5.0	28.4	73.4	**99.3**	0.73	6.7	47.0	57.6	93.4
REFERENCE	100.0	100.0	65.9	73.3	99.7	100.0	100.0	92.8	69.4	99.4

Table 1: Results for automatic evaluation metrics for all systems and domains, using the original continuation as the reference. The metrics are: Length - Average total length per example; Trigrams - % unique trigrams per example; Vocab - % unique words per example.

dynamically during training: As the current samples from the model are used to update the mixture weights, it creates its own learning signal by applying the generative model discriminatively. The SGD learning rate is tuned separately for each dataset.

3.4 Beam Search

Due to the limitations of greedy decoding and the fact that our scoring functions do not decompose across time steps, we perform generation with a beam search procedure, shown in Algorithm 1. The naive approach would be to perform beam search based only on the language model, and then rescore the k best candidate completions with our full model. We found that this approach leads to limited diversity in the beam and therefore cannot exploit the strengths of the full model.

Instead we score the current hypotheses in the beam with the full decoding objective: First, each hypothesis is expanded by selecting the k highest scoring next words according to the language model (we use beam size $k = 10$). Then k sequences are sampled from the k^2 candidates according to the (softmax normalized) distribution over the candidate scores given by the full decoding objective. Sampling is performed in order to increase diversity, using a temperature of 1.8, which was tuned by comparing the coherence of continuations on the validation set.

At each step, the discriminator scores are recomputed for all candidates, with the exception of the entailment score, which is only recomputed for hypotheses which end with a sentence terminating symbol. We terminate beam search when the *termination_score*, the maximum possible score achievable by terminating generation at the current position, is smaller than the current best score.

4 Experiments

4.1 Corpora

We use two English corpora for evaluation. The first is the TripAdvisor corpus (Wang et al., 2010), a collection of hotel reviews with a total of 330 million words.[3] The second is the BookCorpus (Zhu et al., 2015), a 980 million word collection of novels by unpublished authors.[4] In order to train the discriminators, mixing weights, and the SEQ2SEQ and SEQGAN baselines, we segment both corpora into sections of length ten sentences, and use the first 5 sentence as context and the second 5 as the continuation. See supplementary material for further details.

4.2 Baselines

ADAPTIVELM Our first baseline is the same Adaptive Softmax (Grave et al., 2016) language model used as base generator in our framework (§3.1). This enables us to evaluate the effect of our enhanced decoding objective directly. A 100k vocabulary is used and beam search with beam size of 5 is used at decoding time. ADAPTIVELM achieves perplexity of 37.46 and 18.81 on BookCorpus and TripAdvisor respectively.

CACHELM As another LM baseline we include a continuous cache language model (Grave et al., 2017) as implemented by Merity et al. (2018), which recently obtained state-of-the-art perplexity on the Penn Treebank corpus (Marcus et al., 1993). Due to memory constraints, we use a vocabulary size of 50k for CACHELM. To generate, beam search decoding is used with a beam size 5. CACHELM obtains perplexities of 70.9 and 29.71 on BookCorpus and TripAdvisor respectively.

[3] http://times.cs.uiuc.edu/~wang296/Data/

[4] http://yknzhu.wixsite.com/mbweb

BookCorpus	Specific Criteria				Overall Quality		
L2W vs.	Repetition	Contradiction	Relevance	Clarity	Better	Equal	Worse
AdaptiveLM	+0.48	+0.18	+0.12	+0.11	**47%**	20%	32%
CacheLM	+1.61	+0.37	+1.23	+1.21	**86%**	6%	8%
Seq2Seq	+1.01	+0.54	+0.83	+0.83	**72%**	7%	21%
SeqGAN	+0.20	+0.32	+0.61	+0.62	**63%**	20%	17%
LM vs. Reference	-0.10	-0.07	-0.18	-0.10	41%	7 %	**52%**
L2W vs. Reference	+0.49	+0.37	+0.46	+0.55	**53%**	18%	29%

TripAdvisor	Specific Criteria				Overall Quality		
L2W vs.	Repetition	Contradiction	Relevance	Clarity	Better	Equal	Worse
AdaptiveLM	+0.23	-0.02	+0.19	-0.03	**47%**	19%	34%
CacheLM	+1.25	+0.12	+0.94	+0.69	**77%**	9%	14%
Seq2Seq	+0.64	+0.04	+0.50	+0.41	**58%**	12%	30%
SeqGAN	+0.53	+0.01	+0.49	+0.06	**55%**	22%	22%
LM vs. Reference	-0.10	-0.04	-0.15	-0.06	38%	10%	**52%**
L2W vs. Reference	-0.49	-0.36	-0.47	-0.50	25%	18%	**57%**

Table 2: Results of crowd-sourced evaluation on different aspects of the generation quality as well as overall quality judgments. For each sub-criteria we report the average of comparative scores on a scale from -2 to 2. For the overall quality evaluation decisions are aggregated over 3 annotators per example.

Seq2Seq As our evaluation can be framed as sequence-to-sequence transduction, we compare against a seq2seq model directly trained to predict 5 sentence continuations from 5 sentences of context, using the OpenNMT attention-based seq2seq implementation (Klein et al., 2017). Similarly to CacheLM, a 50k vocabulary was used and beam search decoding was performed with a beam size of 5.

SeqGAN Finally, as our use of discriminators is related to Generative Adversarial Networks (GANs), we use SeqGAN (Yu et al., 2017a), a GAN for discrete sequences trained with policy gradients.[5] This model is trained on 10 sentence sequences, which is significantly longer than previous experiments with GANs for text; the vocabulary is restricted to 25k words to make training tractable. Greedy sampling was found to outperform beam search. For implementation details, see the supplementary material.

4.3 Evaluation Setup

We pose the evaluation of our model as the task of generating an appropriate continuation given an initial context. In our open-ended generation setting the continuation is not required to be a specific length, so we require our models and baselines to generate 5-sentence continuations, consistent with the way the discriminator and seq2seq baseline datasets are constructed.

Previous work has reported that automatic measures such as BLEU (Papineni et al., 2002) and Meteor (Denkowski and Lavie, 2010) do not lead to meaningful evaluation when used for long or creative text generation where there can be high variance among acceptable generation outputs (Wiseman et al., 2017; Vedantam et al., 2015). However, we still report these measures as one component of our evaluation. Additionally we report a number of custom metrics which capture important properties of the generated text: *Length* – Average sequence length per example; *Trigrams* – percentage of unique trigrams per example; *Vocab* – percentage of unique words per example. Endings generated by our model and the baselines are compared against the reference endings in the original text. Results are given in Table 1.

For open-ended generation tasks such as our own, human evaluation has been found to be the only reliable measure (Li et al., 2016b; Wiseman et al., 2017). For human evaluation, two possible endings are presented to a human, who assesses the text according to several criteria, which are closely inspired by Grice's Maxims: repetition, contradiction, relevance and clarity. See supplementary material for examples of the evaluation forms we used. For each criterion, the two continuations are compared using a 5-point Likert scale, to which we assign numerical values of -2 to 2. The scale measures whether one generation is strongly or somewhat preferred above the other, or whether they are equal. Finally, the human is asked to make a judgement about overall quality: which ending is better, or are they of equal qual-

[5] We use the implementation available at https://github.com/nhynes/abc.

CONTEXT	We thoroughly enjoyed staying at this hotel. We travelled with our kids aged 4 and 6. The rooms were clean, spacious and well maintained. The beds were great and all of us slept well. The hotel provided vouchers towards breakfast at the adjoining restaurant.
L2W	We didn't have breakfast, so can't comment on that. The concierge was very helpful, and gave us directions, etc. Overall, we would definitely recommend this hotel to others. We would stay here again in a heartbeat. If you're looking for a great location, and good value, I'd recommend this hotel.
ADAPTIVELM	The breakfast buffet was very good, and the staff was very friendly and helpful, and it was a great way to start our day! I would definitely stay here again, and would recommend it to my friends, and would stay there again if I ever go back to Montreal. The location was great, and the staff was friendly and helpful. I would definitely stay here again!!
CACHELM	The staff was very friendly and helpful. The hotel is located in a great location. It is right in the middle of the gaslamp district. I would recommend this hotel to anyone. I would definitely stay here again.
SEQ2SEQ	The staff were very friendly and helpful. The hotel is in a great location. The staff were very friendly and helpful. The hotel is in a great location. The staff were very friendly and helpful.
SEQGAN	We had a breakfast at Shula's & a delicious breakfast. The staff was very helpful and helpful. The breakfast was great as well. The staff was very helpful and friendly. We had a great service and the food was excellent.
REFERENCE	The restaurant was great and we used the vouchers towards whatever breakfast we ordered. The hotel had amazing grounds with a putting golf course that was fun for everyone. The pool was fantastic and we lucked out with great weather. We spent many hours in the pool, lounging, playing shuffleboard and snacking from the attached bar. The happy hour was great perk.

Table 3: Example continuations generated by our model (L2W) and various baselines (all given the same context from TripAdvisor) compared to the reference continuation. For more examples go to https://ari-holtzman.github.io/l2w-demo/.

ity?

The human evaluation is performed on 100 examples selected from the test set of each corpus, for every pair of generators that are compared. We present the examples to workers on Amazon Mechanical Turk, using three annotators for each example. The results are given in Table 2. For the Likert scale, we report the average scores for each criterion, while for the overall quality judgement we simply aggregate votes across all examples.

5 Results and Analysis

5.1 Quantitative Results

The absolute performance of all the evaluated systems on BLEU and Meteor is quite low (Table 1), as expected. However, in relative terms L2W is superior or competitive with all the baselines, of which ADAPTIVELM performs best. In terms of vocabulary and trigram diversity only SEQGAN is competitive with L2W, likely due to the fact that sampling based decoding was used. For generation length only L2W and ADAPTIVELM even approach human levels, with the former better on BookCorpus and the latter on TripAdvisor.

Under the crowd-sourced evaluation (Table 2), on BookCorpus our model is consistently favored over the baselines on all dimensions of comparison. In particular, our model tends to be much less repetitive, while being more clear and relevant than the baselines. ADAPTIVELM is the most competitive baseline owing partially to the robustness of language models and to greater vocabulary coverage through the adaptive softmax. SEQGAN, while failing to achieve strong coherency, is surprisingly diverse, but tended to produce far shorter sentences than the other models. CACHELM has trouble dealing with the complex vocabulary of our domains without the support of either a hierarchical vocabulary structure (as in ADAPTIVELM) or a structured training method (as with SEQGAN), leading to overall poor results. While the SEQ2SEQ model has low conditional perplexity, we found that in practice it is less able to leverage long-distance dependencies than the base language model, producing more generic output. This reflects our need for more complex evaluations for generation, as such models are rarely evaluated under metrics that inspect *characteristics* of the text, rather than ability to predict the gold or overlap with the gold.

For the TripAdvisor corpus, L2W is ranked higher than the baselines on overall quality, as well as on most individual metrics, with the exception that it fails to improve on contradiction and clarity over the ADAPTIVELM (which is again the most competitive baseline). Our model's strongest improvements over the baselines are on repetition and relevance.

Trip Advisor Ablation

Ablation vs. LM	Repetition	Contradiction	Relevance	Clarity	Better	Neither	Worse
REPETITION ONLY	+0.63	+0.30	+0.37	+0.42	**50%**	23%	27%
ENTAILMENT ONLY	+0.01	+0.02	+0.05	-0.10	39%	20%	**41%**
RELEVANCE ONLY	-0.19	+0.09	+0.10	+0.060	36%	22%	**42%**
LEXICAL STYLE ONLY	+0.11	+0.16	+0.20	+0.16	38%	25%	**38%**
ALL	+0.23	-0.02	+0.19	-0.03	**47%**	19%	34%

Table 4: Crowd-sourced ablation evaluation of generations on TripAdvisor. Each ablation uses only one discriminative communication model, and is compared to ADAPTIVELM.

Ablation

To investigate the effect of individual discriminators on the overall performance, we report the results of ablations of our model in Table 4. For each ablation we include only one of the communication modules, and train a single mixture coefficient for combining that module and the language model. The diagonal of Table 4 contains only positive numbers, indicating that each discriminator does help with the purpose it was designed for. Interestingly, most discriminators help with most aspects of writing, but all except repetition fail to actually improve the overall quality over ADAPTIVELM.

The repetition module gives the largest boost by far, consistent with the intuition that many of the deficiencies of RNN as a text generator lie in semantic repetition. The entailment module (which was intended to reduce contradiction) is the weakest, which we hypothesize is the combination of (a) mismatch between training and test data (since the entailment module was trained on SNLI and MultiNLI) and (b) the lack of smoothness in the entailment scorer, whose score could only be updated upon the completion of a sentence.

Crowd Sourcing

Surprisingly, L2W is even preferred over the *original* continuation of the initial text on BookCorpus. Qualitative analysis shows that L2W's continuation is often a straightforward continuation of the original text while the true continuation is more surprising and contains complex references to earlier parts of the book. While many of the issues of automatic metrics (Liu et al., 2016; Novikova et al., 2017) have been alleviated by crowd-sourcing, we found it difficult to incentivize crowd workers to spend significant time on any one datum, forcing them to rely on a shallower understanding of the text.

5.2 Qualitative Analysis

L2W generations are more topical and stylistically coherent with the context than the baselines. Table 3 shows that L2W, ADAPTIVELM, and SEQGAN all start similarly, commenting on the breakfast buffet, as breakfast was mentioned in the last sentence of the context. The language model immediately offers generic compliments about the breakfast and staff, whereas L2W chooses a reasonable but less obvious path, stating that the previously mentioned vouchers were not used. In fact, L2W is the only system not to use the line *"The staff was very friendly and helpful."*, despite this sentence appearing in less than 1% of reviews. The semantics of this sentence, however, is expressed in many different surface forms in the training data (e.g., *"The staff were kind and quick to respond."*).

The CACHELM begins by generating the same over-used sentence and only produce short, generic sentences throughout. Seq2Seq simply repeats sentences that occur often in the training set, repeating one sentence three times and another twice. This indicates that the encoded context is essentially being ignored as the model fails to align the context and continuation.

The SEQGAN system is more detailed, e.g. mentioning a specific location "Shula's" as would be expected given its highly diverse vocabulary (as seen in Table 1). Yet it repeats itself in the first sentence. (e.g. *"had a breakfast", "and a delicious breakfast"*). Consequently SEQGAN quickly devolves into generic language, repeating the incredibly common sentence *"The staff was very helpful and friendly."*, similar to SEQ2SEQ.

The L2W models do not fix every degenerate characteristic of RNNs. The TripAdvisor L2W generation consists of meaningful but mostly disconnected sentences, whereas human text tends to build on previous sentences, as in the reference continuation. Furthermore, while L2W re-

peats itself less than any of our baselines, it still paraphrases itself, albeit more subtly: "we would definitely recommend this hotel to others." compared to "I'd recommend this hotel." This example also exposes a more fine-grained issue: L2W switches from using "we" to using "I" mid-generation. Such subtle distinctions are hard to capture during beam re-ranking and none of our models address the linguistic issues of this subtlety.

6 Related Work

Alternative Decoding Objectives A number of papers have proposed *alternative decoding objectives* for generation (Shao et al., 2017). Li et al. (2016a) proposed a *diversity-promoting objective* that interpolates the conditional probability score with negative marginal or reverse conditional probabilities. Yu et al. (2017b) also incorporate the reverse conditional probability through a noisy channel model in order to alleviate the *explaining-away* problem, but at the cost of significant decoding complexity, making it impractical for paragraph generation. Modified decoding objectives have long been a common practice in statistical machine translation (Koehn et al., 2003; Och, 2003; Watanabe et al., 2007; Chiang et al., 2009) and remain common with neural machine translation, even when an extremely large amount of data is available (Wu et al., 2016). Inspired by all the above approaches, our work presents a general learning framework together with a more comprehensive set of composite communication models.

Pragmatic Communication Models Models for pragmatic reasoning about communicative goals such as Grice's maxims have been proposed in the context of referring expression generation (Frank and Goodman, 2012). Andreas and Klein (2016) proposed a neural model where candidate descriptions are sampled from a generatively trained *speaker*, which are then re-ranked by interpolating the score with that of the *listener*, a discriminator that predicts a distribution over choices given the speaker's description. Similar to our work the generator and discriminator scores are combined to select utterances which follow Grice's maxims. Yu et al. (2017c) proposed a model where the speaker consists of a convolutional encoder and an LSTM decoder, trained with a ranking loss on negative samples in addition to optimizing log-likelihood.

Generative Adversarial Networks GANs (Goodfellow et al., 2014) are another alternative to maximum likelihood estimation for generative models. However, backpropagating through discrete sequences and the inherent instability of the training objective (Che et al., 2017) both present significant challenges. While solutions have been proposed to make it possible to train GANs for language (Che et al., 2017; Yu et al., 2017a) they have not yet been shown to produce high quality long-form text, as our results confirm.

Generation with Long-term Context Several prior works studied paragraph generation using sequence-to-sequence models for image captions (Krause et al., 2017), product reviews (Lipton et al., 2015; Dong et al., 2017), sport reports (Wiseman et al., 2017), and recipes (Kiddon et al., 2016). While these prior works focus on developing neural architectures for learning domain specific discourse patterns, our work proposes a general framework for learning a generator that is more powerful than maximum likelihood decoding from an RNN language model for an arbitrary target domain.

7 Conclusion

We proposed a unified learning framework for the generation of long, coherent texts, which overcomes some of the common limitations of RNNs as text generation models. Our framework learns a decoding objective suitable for generation through a learned combination of sub-models that capture linguistically-motivated qualities of good writing. Human evaluation shows that the quality of the text produced by our model exceeds that of competitive baselines by a large margin.

Acknowledgments

We thank the anonymous reviewers for their insightful feedback and Omer Levy for helpful discussions. This research was supported in part by NSF (IIS-1524371), DARPA CwC through ARO (W911NF-15-1-0543), Samsung AI Research, and gifts by Tencent, Google, and Facebook.

References

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In

Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1173–1182. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Tong Che, Yanran Li, Ruixiang Zhang, R. Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. 2017. Maximum-likelihood augmented discrete generative adversarial networks. *CoRR*, abs/1702.07983.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–226, Boulder, Colorado. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pages 177–190, Berlin, Heidelberg. Springer-Verlag.

Michael Denkowski and Alon Lavie. 2010. Extending the METEOR Machine Translation Evaluation Metric to the Phrase Level. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 250–253.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 623–632, Valencia, Spain. Association for Computational Linguistics.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Edouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. 2016. Efficient softmax approximation for gpus. *arXiv preprint arXiv:1609.04309*.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017. Improving neural language models with a continuous cache. In *International Conference on Learning Representations*.

H Paul Grice, Peter Cole, Jerry Morgan, et al. 1975. Logic and conversation. *1975*, pages 41–58.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *CoRR*, abs/1602.02410.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 329–339.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of the Association of Computational Linguistics*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 48–54. Association for Computational Linguistics.

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Zachary Chase Lipton, Sharad Vikram, and Julian McAuley. 2015. Capturing meaning in product reviews with character-level generative text models. *CoRR*, abs/1511.03683.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike 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. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing lstm language models. *ICLR*.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. 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 Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning (ICML)*, pages 1310–1318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Allen Schmaltz, Alexander M. Rush, and Stuart Shieber. 2016. Word ordering without syntax. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2319–2324, Austin, Texas. Association for Computational Linguistics.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219. Association for Computational Linguistics.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575.

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*.

Hongning Wang, Yue Lu, and ChengXiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *SIGKDD Conference on Knowledge Discovery and Data Mining*.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, Prague, Czech Republic. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017a. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, pages 2852–2858.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomas Kocisky. 2017b. The neural noisy channel. In *International Conference on Learning Representations*.

Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. 2017c. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *arXiv preprint arXiv:1506.06724*.