Retrieval-Augmented Controllable Review Generation

Jihyeok Kim
Yonsei University
zizi1532@yonsei.ac.kr

Seungtaek Choi
Yonsei University
hist0613@yonsei.ac.kr

Reinald Kim Amplayo
University of Edinburgh
reinald.kim@ed.ac.uk

Seung-won Hwang *
Yonsei University
seungwonh@yonsei.ac.kr

Abstract

In this paper, we study review generation given a set of attribute identifiers which are user ID, product ID and rating. This is a difficult subtask of natural language generation since models are limited to the given identifiers, without any specific descriptive information regarding the inputs, when generating the text. The capacity of these models is thus confined and dependent to how well the models can capture vector representations of attributes. We thus propose to additionally leverage references, which are selected from a large pool of texts labeled with one of the attributes, as textual information that enriches inductive biases of given attributes. With these references, we can now pose the problem as an instance of text-to-text generation, which makes the task easier since texts that are syntactically, semantically similar to the output text are provided as inputs. Using this framework, we address issues such as selecting references from a large candidate set without textual context and improving the model complexity for generation. Our experiments show that our models improve over previous approaches on both automatic and human evaluation metrics.

1 Introduction

The ultimate goal of opinion mining and sentiment analysis (Pang and Lee, 2008) is to automatically digest opinions of users towards a certain product to accommodate decision making. While some of these opinions are explicitly articulated in product reviews that users write, most of them are unknown since users have not bought most of the products. Alternative solutions such as aspect-based sentiment analysis (Mukherjee and Liu, 2012; Pontiki et al., 2016) and recommendation systems (Resnick and Varian, 1997; Bobadilla et al., 2013) exist, however these only offer superficial outputs that are not as expressive as textual reviews. Thus, the task of automatically generating reviews given their attributes such as user and product, or *review generation* (Dong et al., 2017), is necessary to achieve this goal.

Most of the previous approaches (Dong et al., 2017; Sharma et al., 2018) have framed review generation as **A2T** (**A**ttribute-to-**T**ext problem), where the given input is a non-linguistic data (*i.e.*, attribute identifiers for user, product, and rating) and the output is the review text. In this problem setup, the key challenge is to learn rich representations of the attributes, which are then used to produce the text using either template-based surface realization methods (Kukich, 1983; McKeown, 1992) or neural-based decoders (Mei et al., 2016; Wiseman et al., 2017), as shown in the red box in Figure 1. However, it is difficult to learn these representations merely from the given attribute identifiers since they do not convey any semantics regarding the attributes.

Our key contribution is **AT2T** (**A**ttribute-matched-**T**ext-to-**T**ext), of augmenting inductive biases of attributes with their matching **reference** reviews, as illustrated as the blue box in Figure 1. For example, as shown in Figure 1, multiple references together contain inductive biases such as frequently reviewed aspects of the product (*e.g.*, talking about plot and character aspects) or habitual user phrases (*e.g.*, "looking forward to the next book"). These references greatly help text generation since not only do they reinforce

*Corresponding author

Proceedings of the 28th International Conference on Computational Linguistics, pages 2284–2295
Barcelona, Spain (Online), December 8-13, 2020

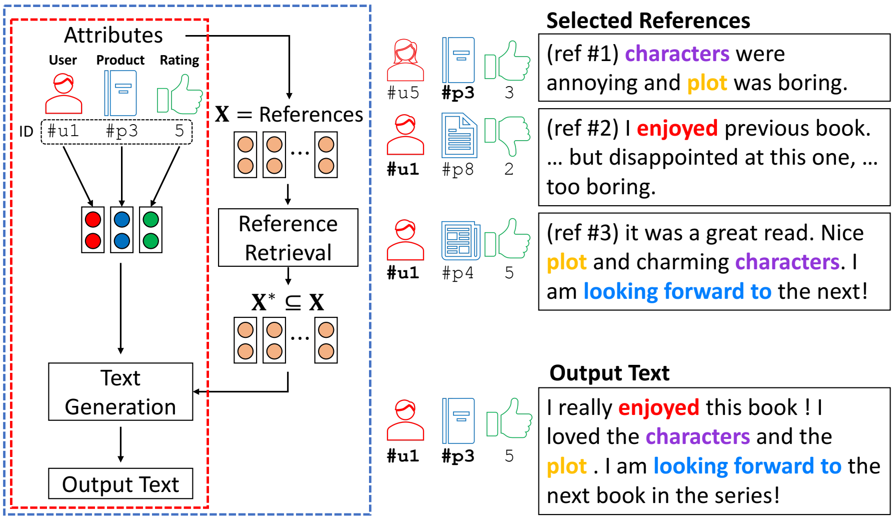
Attributes
User
Product
Rating
ID
#u1
#p3
5
X = References
Reference Retrieval
X* ⊆ X
Text Generation
Output Text
Selected References
#u5
#p3
3
(ref #1) characters were annoying and plot was boring.
#u1
#p8
2
(ref #2) I enjoyed previous book. … but disappointed at this one, … too boring.
#u1
#p4
5
(ref #3) it was a great read. Nice plot and charming characters. I am looking forward to the next!
Output Text
#u1
#p3
5
I really enjoyed this book ! I loved the characters and the plot . I am looking forward to the next book in the series!

Figure 1: High-level diagram of frameworks of previous models (in red box) and our proposed model (in blue box), where we additionally make use of references to generate the output text.

the representations learned from the attributes, they also allow the use of techniques used in sequence-to-sequence learning such as attention (Bahdanau et al., 2015) and copy (See et al., 2017) mechanisms. In related problem domains of generating abstractive summaries or dialogue utterances, such bias is introduced by a **T2T** (**T**ext-to-**T**ext) approach, of generating an extractive summary first (Gehrmann et al., 2018) or retrieving informative prior turns (Cai et al., 2018), then generating final outputs using these as references. Central to the framework is reference retrieval, since relevant references provide valuable context, but, in contrast, noisy references rather hinder generation.

For reference retrieval in T2T, lexical features, *e.g.*, TF-IDF, have been used, assigning relevance based on the degree of word overlap between two texts. In AT2T, however, lexical features are not directly applicable or effective. **First,** unlike T2T where input and output are both texts, our input is a list of identifiers, *i.e.*, (user ID, product ID, rating). As a result, we cannot expedite the process of finding matching references, as in T2T solutions using lexical features for fast retrieval, *e.g.*, using inverted index. **Second,** lexical similarity cannot fully capture rating, as sentiment lexicons appear in a small portion of text (Li et al., 2018). For example, flipping a single lexicon (from 'good' to 'bad') from lexically identical sentences can completely invert the rating. One alternative solution, complementing lexical similarity, is to assign additional credits to references labeled with the input rating. However, references – labeled with a different rating, but having useful rating-related context – are forced to be penalized. For example, in Figure 1, no additional credits are assigned to ref#2 labeled with the nearly opposite rating, although it contains useful context for the given rating, *e.g.*, "enjoyed". We later empirically show that neither lexical similarity nor rating accuracy of references guarantee rating semantics of generated reviews.

To address these limitations, we propose two approaches: **pseudo-supervised** and **reinforcement** learning framework, denoted as SL and RL respectively. **First,** we expedite matching in SL using identifiers. For efficient retrieval without lexical features, we propose a parametric coarse-filtering approach using attribute identifiers, having constant time complexity for each instance in a candidate pool. **Second,** to generate reviews which are compatible with input rating, we retrieve references which maximize the rating accuracy of generated reviews - rather than references labeled with the input rating. RL enables such retrieval: a retrieval model is trained to maximize rewards including rating accuracy as well as lexical similarity of generated reviews.

To validate the effectiveness of AT2T, we perform experiments on a dataset consisting of product reviews from Amazon Books, aligned with their corresponding attributes: user, product and rating (Dong et al., 2017). Our experiments using automatic evaluations show that utilizing relevant references hugely helps generation in terms of content similarity, and rating accuracy. Moreover, our human evaluations show that our model generates more informative and grammatical texts compared to previous models.

2 Related Work

Data-to-Text Generation Our task is generally related to a suite of tasks on data-to-text (D2T) generation, where database tables (Wiseman et al., 2017), RDF graphs (Belz et al., 2011), and knowledge base relations (Perez-Beltrachini et al., 2016) are explored as inputs. A variety of neural-based models have been used on these tasks, including vanilla sequence-to-sequence models (Mei et al., 2016), extended by explicitly incorporating context selection and planning (Puduppully et al., 2019a), by employing graph-based neural networks (Marcheggiani and Perez-Beltrachini, 2018), and by modeling entities (Puduppully et al., 2019b). While review generation is essentially a subtask of D2T, it is relatively understudied than other D2T tasks. Previous models include an encoder-decoder model with attention (Dong et al., 2017), improved by including an objective function for rating accuracy (Sharma et al., 2018; Li and Tuzhilin, 2019), by introducing a hierarchical decoder (Zang and Wan, 2017), by decomposing the decoding stage as coarse-to-fine manner (Li et al., 2019), and by using additional inputs such as user-given summary (Ni and McAuley, 2018) or product description (Li and Tuzhilin, 2019). In this paper, we make performance improvements by proposing a concept of leveraging references, and extensions proposed in the aforementioned literature are orthogonal and thus applicable to improve our models further.

Augmenting context using references While data-hungry neural models for some task may afford sufficient training resources, some other tasks such as sentence-level classification (Kim, 2014) and summarization (Rush et al., 2015) suffer from limited context, given a single sentence as context. Review generation can be viewed as an extreme case of limited context, totally lacking textual context and thus depending solely on a small set of attribute identifiers as input.

For text classification tasks, solutions have been to increase the context, by adding inherent and induced metadata such as topics (Zhao and Mao, 2017) and translations (Amplayo et al., 2018). Meanwhile references have been used as additional source to augment context in text-to-text generation tasks such as summarization (Cao et al., 2018; Peng et al., 2019), machine translation (Gu et al., 2018), or dialogue system (Song et al., 2018; Pandey et al., 2018; Weston et al., 2018; Zhu et al., 2019). References can be seen as a new and effective additional context that introduces inductive biases of attributes that can only be found in texts. However, retrieval task is much harder in our task than in previous tasks, as we have inputs of attribute identifiers having little information for retrieval.

3 AT2T

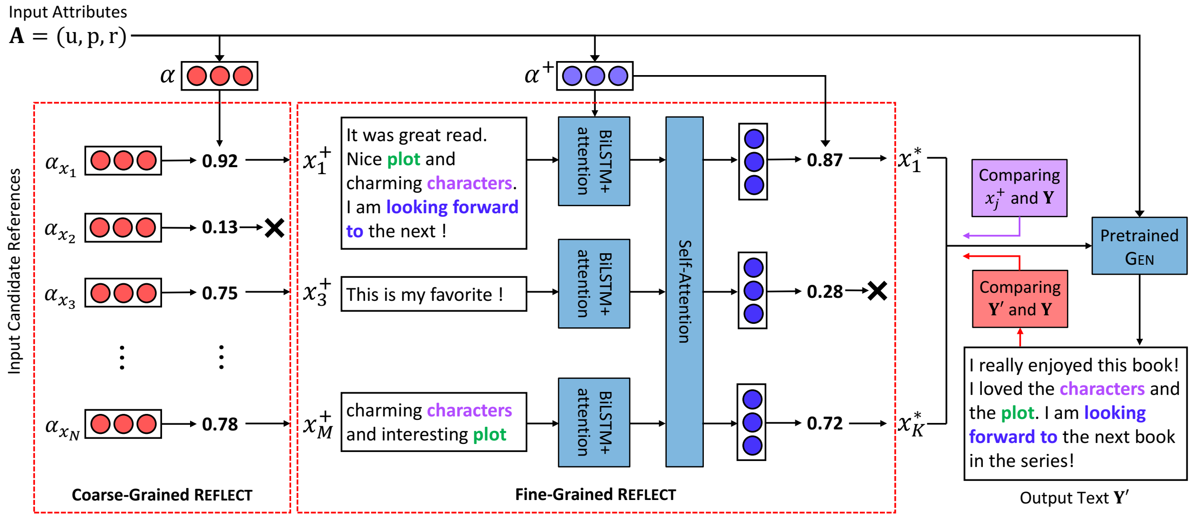
Input Attributes
$\mathbf{A} = (u, p, r)$
α
α^+
Input Candidate References
α_{x_1}
0.92
α_{x_2}
0.13
α_{x_3}
0.75
α_{x_N}
0.78
Coarse-Grained REFLECT
x_1^+
It was great read. Nice plot and charming characters. I am looking forward to the next !
x_3^+
This is my favorite !
x_M^+
charming characters and interesting plot
BiLSTM+ attention
Self-Attention
0.87
0.28
0.72
x_1^*
x_K^*
Fine-Grained REFLECT
Comparing x_j^+ and $\mathbf{Y}$
Comparing $\mathbf{Y}'$ and $\mathbf{Y}$
Pretrained GEN
I really enjoyed this book! I loved the characters and the plot. I am looking forward to the next book in the series!
Output Text $\mathbf{Y}'$

Figure 2: The full architecture of REFLECT when integrated to a generation model, which consists of three components: Coarse-Grained REFLECT, Fine-Grained REFLECT, and the pretrained GEN model. We use different encoders for α and α^+ with the same identifiers $\mathbf{A}$.

This work studies review generation task, where we are given review-specific attributes such as user, product, and rating, *i.e.*, $\mathbf{A} = \{u, p, r\}$ as input and the corresponding review $\mathbf{Y} = \{y_i\}_{i=1}^{L}$ as output.

We reformulate the problem by introducing references. In review domain, references are reviews from the training dataset that are either written by user u or written for product p. That is, references can be reviews from another user but from the same product, *i.e.*, (u', p) where $u' \neq u$, and vice versa[1]. This introduces an additional input to our text generation model: a set of N references $\mathbf{X} = \{x_j\}_{j=1}^{N}$, where $x_j = \{w_i\}_{i=1}^{L_j}$ is the jth reference with L_j tokens. Through this reformulation (AT2T), we can now pose the task as text-to-text generation, where we generate an output given $\mathbf{X}$ and $\mathbf{A}$ as inputs.

The new problem setting introduces a major challenge since there can be a large number of references. As most of the references are irrelevant, efficiently and effectively selecting the optimal K references $\mathbf{X}^* = \{x_1^*, ..., x_K^*\} \subseteq \mathbf{X}$ is one of the essential sub-tasks to AT2T. To this end, in AT2T, models consist of (1) **ref**erence se**lect**ion modules (REFLECT) where we select relevant references $\mathbf{X}^*$ to given attributes $\mathbf{A}$ and (2) a **gen**eration module (GEN) of utilizing references as inductive biases of the attributes.

While, in T2T tasks, lexical features have been used for retrieval, those cannot be directly applied for AT2T, having inputs of identifiers without text contents. To overcome such difficulty, we propose two approaches on retrieval stage with different learning schemes which are pseudo-supervised learning (SL) and reinforcement learning (RL). We first introduce a SL method to construct trainable REFLECT without text contents as inputs. Then, we propose to train REFLECT using RL where we enable the model to effectively preserve rating semantics. We show an overview of our approach in Figure 2.

4 SL-REFLECT

Given the attributes $\mathbf{A}$ and reference candidates $\mathbf{X}$, REFLECT selects the most relevant K references $\mathbf{X}^* = \{x_1^*, ..., x_K^*\}$ from $\mathbf{X}$. In T2T tasks having texts as inputs, text-based, non-parametric matching has been used for retrieval, *e.g.*, TF-IDF. However, given only identifiers, we cannot use such matching. In this section, we explore a pseudo-supervised learning approach to train parametric retrieval models.

Relevance Pseudo-Supervision In contrast to T2T tasks, references should contain relevant contexts to the input rating, to guide GEN to generate a rating-consistent review. Although references largely comprising of words presented in target review are useful for providing overall contents, lexical similarity does not guarantee semantic relevance especially for rating, *e.g.*, two identical sentences with one word difference such as "good" or "bad" have significant rating differences.

To consider rating information, we propose to generate a pseudo-label z_{x_j} for each reference candidate x_j. We use rating accuracy and lexical similarity which are linearly combined by λ:

$$z_{x_j} = (1 - \lambda) * \text{LEXSIM}(x_j, \mathbf{Y}) + \lambda * I(r = r_j), \quad (1)$$

where LEXSIM denotes lexical similarity between each reference candidate x_j and a ground-truth review $\mathbf{Y}$, $I(r = r_j)$ is an indicator variable for rating accuracy (1 for the same rating, 0 otherwise), and λ is a balancing factor between lexical similarity and rating accuracy. We adopt average of uni-/bi-/quad-gram BLEUs (Papineni et al., 2002) as LEXSIM which was effective in our experiments. We train SL-REFLECT models using binary cross entropy as objective and z_{x_j} as supervision. Note that, the gold review $\mathbf{Y}$ is only needed to provide supervision during training, and is not required for inference.

Coarse-Grained REFLECT To maximize computational efficiency, previous approaches in T2T coarsely retrieve the small number of promising references, $\mathbf{X}^+ = \{x_1^+, ..., x_M^+\} \subseteq \mathbf{X}$, using efficient matching such as TF-IDF or inverted index, then rerank using more effective but expensive matching to select best K references, $\mathbf{X}^* \subseteq \mathbf{X}^+$ where $K < M \ll N$.

Instead of text-based matching which is not available in our task, we propose to use attribute-based parametric matching, which has $O(N)$ time complexity and is fully parallelizable. More formally, we match input attributes $\mathbf{A}$ with attributes of candidates $\mathbf{A}^{\mathbf{X}} = \{\mathbf{A}^{x_1}, ..., \mathbf{A}^{x_N}\}$, where

[1] Though sharing the same rating can also be a criteria for reference candidates, we empirically found this would increase the candidate size too much, while we can apply rating bias by including rating accuracy in relevance supervision instead.

$\mathbf{A}^{x_j} = \{u^{x_j}, p^{x_j}, r^{x_j}\}$. We encode attribute features using embedding matrices followed by a fully connect layer, and calculate relevance score using inner product between them. We denote concatenation of the attribute vectors for the input identifiers, $u/p/r$, by $\mathbf{a} \in \mathbb{R}^{3\times d_c}$ where d_c is the number of features in vectors. For simplicity, we use different superscripts for $\mathbf{a}$ to indicate that different embedding matrices are used with the same identifiers, and use different subscripts to indicate different identifiers.

$$\alpha_{x_j} = \tanh(\mathbf{H}_x \mathbf{a}_{x_j}) \in \mathbb{R}^{d_c} \tag{2}$$

$$\alpha = \tanh(\mathbf{H}\mathbf{a}) \in \mathbb{R}^{d_c} \tag{3}$$

$$s(x_j) = \sigma(\alpha_{x_j}^{\top}\alpha) \in (0,1), \tag{4}$$

where $\mathbf{a}_{x_j}$ and $\mathbf{a}$ are attribute vectors for $\mathbf{A}^{x_j}$ and $\mathbf{A}$ respectively, $\mathbf{H}_x$ and $\mathbf{H}$ are learnable matrices, and σ denotes sigmoid function. For $\mathbf{a}_{x_j}$ and $\mathbf{a}$, we share the same embedding matrices for efficient training.

Fine-Grained REFLECT Now that we narrowed down to $M \ll N$ reference candidates, we can afford to use expensive textual input features. Fine-Grained REFLECT accepts the references $\mathbf{X}^+$ as input and outputs a score for each candidate that is used to select the final K references.

To get document-level encoding d_j^+ for each candidate x_j^+, we use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) with attention pooling (Bahdanau et al., 2015) using $\mathbf{A}$ as attention query.

$$\alpha^+ = \tanh(\mathbf{M}\mathbf{a}^+) \in \mathbb{R}^{d_c} \tag{5}$$

$$\mathbf{h}_j^+ = \text{BiLSTM}(\{\mathbf{w}_{jk}^+\}_{k=1}^{L_j^+}) \in \mathbb{R}^{L_j^+ \times d_f} \tag{6}$$

$$\mathbf{d}_j^+ = \text{softmax}_k(\mathbf{v}_a^{\top} \tanh(\mathbf{H}_a[\mathbf{h}_{jk}^+; \alpha^+])\mathbf{h}_j^+ \in \mathbb{R}^{d_f}, \tag{7}$$

where $\mathbf{a}^+$ is obtained from another embedding matrix using $\mathbf{A}$ as identifiers, $\mathbf{M}$ and $\mathbf{H}_a$ are learnable matrices and $\mathbf{v}_a \in \mathbb{R}^{d_f}$ is a learable vector. Note that, we use different parameters for α in Equation 3 and α^+ in Equation 5 since different features are required for Coarse-Grained REFLECT and Fine-Grained REFLECT: features for attribute-attribute matching and features for attribute-text matching respectively.

We also add a self-attention layer (Vaswani et al., 2017) to further contextualize reference $\{\mathbf{d}_j^+\}_{j=1}^{M}$.

$$[\mathbf{q};\mathbf{k};\mathbf{v}]_j = [\mathbf{W}_q;\mathbf{W}_k;\mathbf{W}_v]\mathbf{d}_j^+ \in \mathbb{R}^{d_f} \tag{8}$$

$$\tilde{\mathbf{d}}_j^+ = f_{\text{FF}}\left(\text{softmax}\left(\frac{\mathbf{q}^{\top}\mathbf{k}}{\sqrt{d_f}}\right)\mathbf{v}\right) \in \mathbb{R}^{M\times d_f}, \tag{9}$$

where f_{FF} is a residual feed-forward layer. Cross-reference contextualization further clarifies the meaning of each reference by considering latent dependency over references (Liu and Lapata, 2019).

Then, we estimate relevance score using inner product between $\tilde{\mathbf{d}}_j^+$ and α^+, i.e., $s(x_j^+) = \sigma(\tilde{\mathbf{d}}_j^{+\top}\alpha^+)$.

5 RL-REFLECT

SL has the advantage of efficient selection of references by lexical and rating similarity, but this cannot guarantee generated documents would preserve such similarity. In addition, "selection" of reference can be generalized into "composing" multiple documents. We first concretely describe the motivation for introducing RL-REFLECT using the example presented in Figure 1.

Pseudo-labels evaluate the lexical and rating similarity of the entire reference document, and encourage to select ref#3 in Figure 1. However, in a hypothetical scenario without ref#3, ref#1 and ref#2 can be "composed", with the former contributing to aspects to be discussed (*e.g.*, "character" and "plot") and the latter to rating (*e.g.*, "enjoyed"), while pseudo-labels may underestimate the importance of both. Especially with respect to the goal of attaining rating semantics, pseudo-labels discourage REFLECT to retrieve ref#2, with nearly opposite rating to the ground truth, though other aspects can guide the generation strongly.

Instead, to compute the aggregated effect of partial contributions, we employ Reinforcement Learning (RL): We first sample a set of references, generate a review using the references, and then estimate partial relevance of those by how those are reflected in the generation. As illustrated with the above example, RL is better than SL, especially in scenarios where multiple references collaboratively contribute to the generation, while SL is comparable if some document is dominant in all aspects.

Specifically, training retrieval models using RL involves (1) sampling a set of references $\mathbf{X}^*$ from a candidate pool; (2) generating a review $\mathbf{Y}'$ using GEN with $\mathbf{X}^*$; (3) calculating a reward for $\mathbf{X}^*$ based on $\mathbf{Y}$ and $\mathbf{Y}'$; and (4) adapting the sampling towards the direction to maximize the reward. We set reward function by replacing x_j with $\mathbf{Y}'$ in the previously defined formula on relevance score (Equation 1). To obtain rating accuracy of $\mathbf{Y}'$, we first train a standard sentiment classifier comprising of a bidirectional LSTM layer followed by an attention pooling layer, and then predict rating of $\mathbf{Y}'$ using the classifier. For efficient RL training, we use filtered references by Coarse-Grained REFLECT, *i.e.*, $\mathbf{X}^+$, as candidates.

A challenging part is adapting the sampling $\mathbf{X}^*$ (the 4th step described above) during training, since sampling is a discrete operation which breaks continuity of a function and thus violates differentiability assumption in standard backpropagation algorithm. Likelihood-ratio trick, proposed in (Williams, 1992), enables the backpropagation of gradients regardless of discrete sampling.

$$\nabla J = \mathbb{E}_\tau[\nabla \log p(\mathbf{X}^+) z_{\mathbf{Y}'}] \tag{10}$$

$$\approx \frac{1}{B}\sum_{b=1}^{B} \nabla \log p(\mathbf{X}_b^+) z_{\mathbf{Y}'}^b \tag{11}$$

where $p(\mathbf{X}^+)$ is probability distribution of selecting $\{x_j^+\}_{j=1}^M$ which is obtained by normalizing relevance scores, *i.e.*, $p(x_j^+) = s(x_j^+)/\sum_{m=1}^M s(x_m^+)$, and B is the number of sampling trials for approximation.

Stabilizing Training of RL Despite the above-mentioned strength of adapting to a heuristic scoring function for generation, RL training is notoriously unstable and converges slowly (Ranzato et al., 2015). Our key contribution is to make RL training practical. Note, we keep the weights of pretrained GEN fixed to keep RL training cost low. First, we use BLEU-1 as a proxy of look-ahead exploration. Compared to random exploration for all references increasing variance too much, we prioritize exploration to those with high lexical similarity (*e.g.*, BLEU-1 no less than 0.2). We empirically found this prioritization reduces the variance without compromising the reference quality much. Second, as greedily maximizing for R induces high variance, we maximize $R - \tilde{R}$ instead of R, where $\tilde{R}$ is a sub-optimal reward obtained by a baseline, specifically, $p(\mathbf{X}^+)$ (Dong et al., 2018). Introducing a baseline performance is known to decrease the variance (Weaver and Tao, 2001), and our experience was consistent.

6 GEN

GEN follows an encoder-decoder framework equipped with copying mechanism. First, we encode each reference x_j^* with L_j^* words using a bidirectional LSTM with attention pooling.

$$\mathbf{h}_j^* = \text{BiLSTM}(x_j^*) \in \mathbb{R}^{L_j^* \times d_g^e} \tag{12}$$

$$\mathbf{d}_{tj}^* = \text{softmax}_k(\mathbf{v}_{g_w}^\top \tanh(\mathbf{H}_{g_w}[\mathbf{h}_{jk}^*; \mathbf{a}^{g_w}; \mathbf{o}_t])\mathbf{h}_j^*, \tag{13}$$

where $\mathbf{v}_{g_w}, \mathbf{h}_{jk}^*, \mathbf{d}_{tj}^* \in \mathbb{R}^{d_g^e}$ and $\mathbf{o}_t \in \mathbb{R}^{d_g^h}$ is a hidden state of decoder at t-th generation.

As in Fine-Grained REFLECT (Equation 8, 9), we further contextualize reference encodings, $\mathbf{d}_{tj}^*$, using a self-attention layer, yielding $\tilde{\mathbf{d}}_{tj}^*$. Then, we aggregate $\tilde{\mathbf{d}}_{tj}^*$ using another attention layer.

$$\mathbf{o}_t^{\text{ref}} = \text{softmax}(\mathbf{v}_{g_x}^\top \tanh(\mathbf{H}_{g_x}[\tilde{\mathbf{d}}_{tj}^*; \mathbf{a}^{g_x}; \mathbf{o}_t])\tilde{\mathbf{d}}_t^*, \tag{14}$$

where $\mathbf{v}_{g_x}, \mathbf{o}_t^{\text{ref}} \in \mathbb{R}^{d_g^e}$ and $\tilde{\mathbf{d}}_t^* \in \mathbb{R}^{K \times d_g^e}$. Similar to encoding references, we attend over embedding vectors of input attributes, $\mathbf{a}^{\text{emb}} \in \mathbb{R}^{3 \times d_g^a}$, yielding $\mathbf{o}_t^{\text{emb}} \in \mathbb{R}^{d_g^a}$.

For decoder, we use a two-layer LSTM, and to obtain initial hidden state of LSTM, we use a multilayer perceptron following (Dong et al., 2017) taking attribute vectors as input.

$$\mathbf{h}_0 = \tanh(\mathbf{W}\mathbf{a}^{\text{emb}}) \in \mathbb{R}^{d_g^h} \tag{15}$$

For each generation step, decoder uses both reference vectors $\mathbf{o}_t^{\text{ref}}$ and attribute vectors $\mathbf{o}_t^{\text{emb}}$.

$$\mathbf{o}_t = \text{2-layer-LSTM}(\mathbf{o}_{t-1}, \mathbf{y}_t) \in \mathbb{R}^{d_g^h} \tag{16}$$

$$\mathbf{s}_t = \tanh(\mathbf{W}_o[\mathbf{o}_t^{\text{ref}}; \mathbf{o}_t^{\text{emb}}; \mathbf{h}_t]) \in \mathbb{R}^{d_g^h} \tag{17}$$

$$\mathbf{g}_t = \text{softmax}(\mathbf{H}_s \mathbf{s}_t), \tag{18}$$

where y_t is generated word at step t.

As habitual words/phrases of users/products can be reused for generation, we allow GEN to copy words from references.

$$\mathbf{z}_t = \sigma(\mathbf{W}_z[\mathbf{y}_t; \mathbf{h}_t; \mathbf{o}_t^{\text{emb}}; \mathbf{o}_t^{\text{ref}}]) \in (0, 1) \tag{19}$$

$$\mathbf{p}_t = \mathbf{z}_t \times \mathbf{g}_t + (1 - \mathbf{z}_t) \times \text{att}_t, \tag{20}$$

where att_t is attention score distribution for words in references at Equation 13. For inference, we greedily generate words by argmax-ing $\mathbf{p}_t$.

Training GEN We pretrain GEN using top-K BLEU-1 score references as inputs and cross-entropy on gold reviews as the loss function. When we trained GEN using retrieved references by REFLECT, performance dropped significantly. We suspect this is because even a small amount of irrelevant references misguide training of GEN.

7 Experiments

We used the same dataset used in (Dong et al., 2017) (Amazon Book reviews) to evaluate models. Each instance in the dataset consists of a review and aligned attributes (user ID, product ID, and a rating ranging from 1 to 5). Statistics of the number of references for each attribute are as follows: (1) minimum number of references is 6, 2, 13K for user, product, rating respectively, (2) maximum number of references is 1265, 351, 405K, and (3) average number of references is 33.34, 8.17, 131K.

7.1 Training Details

For efficient hyperparameter search on the vector dimension, we used 64 for attribute vectors (i.e., d_c, d_g^a) following (Dong et al., 2017), and for others we choose the best value among 128, 256, and 512 using validation set, where d_f, d_g^e, d_g^h were 256, 128, and 512 respectively. Word embedding matrices for Fine-Grained REFLECT and GEN were pre-trained using fastText (Bojanowski et al., 2017). For the number of references, we set M and K to be 50 and 10 respectively.

For training of GEN, we used the same setting with (Dong et al., 2017) including batch size, optimizer, learning rate scheduling, initialization of parameters, dropout ratio, and gradient clipping. We excluded references having BLEU-1 score to ground-truth review less than 0.2. If all reference candidates are excluded, we set $\mathbf{o}_t^{\text{ref}}$ as zero vector. For training of SL-REFLECT, we set batch size to be 50,000 and 150 for Coarse- and Fine-Grained REFLECT respectively, and use Adam (Kingma and Ba, 2015) optimizer with learning rate 0.001 for both models. For training of RL-REFLECT, we set both the number of samples B and batch size to be 50, and λ to be 0.04. We searched the optimal λ among $[0.0, 0.02, 0.04, 0.06, 0.08, 0.1]$. We used Adam optimizer with learning rate 0.0001. Our experiments were conducted on a GTX-2080Ti GPU.

7.2 Evaluation Results

Models As baselines, we report performance of Attr2Seq (Dong et al., 2017) and Cyclegen (Sharma et al., 2018) which use embedding vectors to encode given attributes. For our models, we report performance of GEN using retrieved references by Coarse- and Fine-Grained REFLECT trained using SL and RL, denoted by GEN-C-F (SL) and GEN-C-F (RL) respectively[2]. In addition, we also report performances of RETRIEVE-C-F by evaluating the top-1 retrieved reference.

[2] In our preliminary evaluation on GEN-C-F (SL→RL) where we first pretrained REFLECT using SL and fine-tuned using RL and GEN-C-F (SL+RL) where we train REFLECT by jointly optimizing both SL supervision and RL rewards, the models performed poorly, so we do not consider such combination.

Model		Lexical Sim	Phrase Sim	Rating Accuracy
		BLEU-1	BLEU-4	
Baseline	Attr2Seq	30.48	5.03	-
	Attr2Seq†	28.57	4.74	76.89
	Cyclegen	30.63	5.46	-
Ours	Retrieve-C-F	30.28	4.85	80.33
	Gen-C-F (SL)	31.82	**5.65**	78.93
	Gen-C-F (RL)	**32.03**	5.58	**84.48**

Table 1: BLEU and rating accuracy results on test dataset. We report performances of Attr2Seq model reported in the original paper, and of our best-effort implementation for unreported measures (denoted by Attr2Seq†). **Bold** denotes the best result, and underlined denotes results outperforming all baselines.

Model	**Output**
Input Attributes	*User*: hooked, men, family, love, **stories**, Mackenzie, **Hart**, series, Lilianna *Product*: **characters**, looking, next, **story**, believable, fast, author, mystery, moving *Rating*: 1
Gold	**Hart** is a good author but I found this series short on content and **characters** lacking. Length and depth of **story** left you wanting more.
Attr2Seq†	I am not sure what to say about this book. I am not sure what I was expecting but I was disappointed.
Retrieve-C-F	A chicks club with a difference, good fast moving **story** with believable **characters**. I will read more of this author and looking forward to the next book to this series.
Gen-C-F (SL)	The author has a great **story** line and the **characters** are great. I will read more of her books. I will read more of her work.
Gen-C-F (RL)	The beginning of the book was great, but the last “num”% of the book was boring and the **characters** were boring. I didn’t like the **story** and the **characters**.

Table 2: Example generated outputs from four different systems. **Bold-faced** tokens are words that are also found in the attribute-specific-frequent words. Words colored blue and red contain consistent and inconsistent rating information to the given rating respectively.

Metrics We validate models based on two criteria and corresponding metrics as follows: (1) *Content similarity* between generated reviews and ground-truth reviews can be measured by widely adapted metric, BLEU (Papineni et al., 2002). (2) We measure *rating accuracy* via classification accuracy of generated reviews using pretrained rating classifier[3].

Automatic Evaluation This section compares our model with existing baselines based on content similarity and rating accuracy. Results are presented in Table 1.

Retrieve-C-F show worse performance compared to generation approaches including baselines, except for rating accuracy. This is because aspects or opinions to be discussed can be flexible depending on input attributes, but retrieval approaches are limited to predicting existing reviews. On the other hand, Gen-C-F (RL) and Gen-C-F (SL) utilizing existing reviews as references significantly outperform all baselines. This validates our hypothesis that inductive biases improve generation performance, and our Reflect models can retrieve helpful references.

Human Evaluation We also conducted human evaluations using Amazon Mechanical Turk system to evaluate 150 randomly sampled texts. We compared a retrieval model, Retrieve-C-F, and three generation models including Attr2Seq†, Gen-C-F (SL), and Gen-C-F (RL).

For each attribute pair, participants were asked to blindly compare outputs of the four models and annotate the best and worst. Specifically, three participants were shown each paired four outputs as well as corresponding ground-truth text and were asked to decide which is the best and the worst according to three criteria such as *Informativeness* (Does the review convey all information found in the gold standard review?), *Correctness* (Is the information in the review factually accurate based on the information in the

[3] We use a BiLSTM with an attention layer for the rating classifier, which is trained using Amazon review dataset and optimized using cross entropy as objective function

gold standard review?), and *Compactness* (Is the review written in a complete yet concise manner?).

We can then observe whether any model is significantly better in terms of Best-Worst Scaling (Louviere and Woodworth, 1991) where the percentage of times it was selected as best subtracted by the percentage of times it was selected as worst. In Table 3, we can observe that GEN-C-F (RL) significantly outperforms Attr2Seq† as well as RETRIEVE-C-F) at $p < 0.05$ using t-test.

Model		Best-Worst
Baseline	Attr2Seq†	-8.89%
Ours	RETRIEVE-C-F	-2.22%
	GEN-C-F (SL)	-0.67%
	GEN-C-F (RL)	**11.78%**

Table 3: Human evaluation result.

Qualitative Example Table 2 shows examples of generated reviews from different models with given attributes for the given attributes. We also present frequently used words of the user and the product which are gathered by ranking words using TF-IDF.

As can be seen in the examples, Attr2Seq† tends to generate generic reviews, where the generated words do not coincide with the attribute-specific-frequent words. RETRIEVE-C-F is able to generate attribute-specific-frequent words such as *characters* and *story*, however it also outputs inconsistent information where rating semantic does not match to the given rating. Utilizing references as inputs, GEN-C-F (SL) and -F (RL) are also able to generate attribute-specific terms. For rating accuracy, while GEN-C-F (SL) generates words that are inconsistent with the given rating, such as *great*, GEN-C-F (RL), optimized to increase rating accuracy of generated reviews, is able to preserve rating accuracy.

Analysis on Rating Consistency We analyze contribution of rating accuracy for relevance of references in Equation 1 on overall quality and rating consistency of generated reviews, where BLEU-1/4 and rating accuracy are adopted as evaluation metrics. We show results on Table 4.

Introducing rating accuracy for relevance, in addition to lexical similarity, increased rating accuracy of generated reviews, and even BLEU-1 and BLEU-4. Without rating accuracy, *i.e.*, $\lambda = 0$ in Equation 1, both GEN-C-F (SL) and GEN-C-F (RL) produced lower rating accuracy of generated reviews than that of Attr2Seq†. This shows that lexical similarity may fail to capture rating semantics. Meanwhile, GEN-C-F (RL) significantly outperformed GEN-C-F (SL) on rating accuracy with little sacrifice of BLEU-4, which indicates that relevance of references regarding rating semantics should be determined based on generated reviews rather than on references.

Model	BLEU-1	BLEU-4	RA
Attr2Seq†	28.57	4.74	76.89
GEN-C-F (SL)	31.82	**5.65**	78.93
- RA	31.43	5.60	75.49
GEN-C-F (RL)	**32.03**	5.58	**84.48**
- RA	31.95	5.53	74.75

Table 4: Ablation on relevance estimation. "RA" denotes rating accuracy and "-RA" indicates $\lambda = 0$ in Equation 1. We **bold** the best performance on each metric. Values colored red are performances weaker than that of Attr2Seq†.

8 Conclusion

In this work, we study the problem of review generation guided by reference documents. Attribute-specific reference reviews provide useful inductive biases of given attributes, and such biases can be explicitly delivered to generated reviews using copying mechanism. Inspired by promising results, as future work, we will investigate alternative means of guidance, such as keywords or topic distribution of attributes.

Acknowledgements

This work was supported by AI Graduate School Program (2020-0-01361) and by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Resarch Center) support program (IITP-2020-2016-0-00464) supervised by the IITP (Institute for Information Communications Technology Planning Evaluation).

References

Reinald Kim Amplayo, Kyungjae Lee, Jinyoung Yeo, and Seung-won Hwang. 2018. Translations as additional contexts for sentence classification. In *IJCAI*, pages 3955–3961.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Anja Belz, Mike White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first surface realisation shared task: Overview and evaluation results. In *ENLG*, pages 217–226.

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems*, 46:109–132.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Deng Cai, Yan Wang, Victoria Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2018. Skeleton-to-response: Dialogue generation guided by retrieval memory. *arXiv preprint arXiv:1809.05296*.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *ACL*, pages 152–161.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *EACL*, pages 623–632.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. Banditsum: Extractive summarization as a contextual bandit. In *EMNLP*, pages 3739–3748.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *ACL*, pages 145–150.

Pan Li and Alexander Tuzhilin. 2019. Towards controllable and personalized review generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3237–3245, Hong Kong, China, November. Association for Computational Linguistics.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *NAACL-HLT*, pages 1865–1874.

Junyi Li, Wayne Xin Zhao, Ji-Rong Wen, and Yang Song. 2019. Generating long and informative reviews with aspect-aware coarse-to-fine decoding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1969–1979, Florence, Italy, July. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *ACL*, pages 5070–5081.

Jordan J Louviere and George G Woodworth. 1991. Best-worst scaling: A model for the largest difference judgments. *University of Alberta: Working Paper*.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *INLG*, pages 1–9.

Kathleen R. McKeown. 1992. *Text generation - using discourse strategies and focus constraints to generate natural language text*. Studies in natural language processing. Cambridge University Press.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *NAACL-HLT*, pages 720–730.

Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 339–348, Jeju Island, Korea, July. Association for Computational Linguistics.

Jianmo Ni and Julian McAuley. 2018. Personalized review generation by expanding phrases and attending on aspect-aware representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 706–711, Melbourne, Australia, July. Association for Computational Linguistics.

Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar encoder-decoder for neural conversation generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1329–1338.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Hao Peng, Ankur P. Parikh, Manaal Faruqui, Bhuwan Dhingra, and Dipanjan Das. 2019. Text generation with exemplar-based adaptive decoding. In *NAACL-HLT*, pages 2555–2565.

Laura Perez-Beltrachini, Rania Sayed, and Claire Gardent. 2016. Building RDF content for data-to-text generation. In *COLING*, pages 1493–1502.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California, June. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019a. Data-to-text generation with content selection and planning. In *AAAI*, pages 6908–6915.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019b. Data-to-text generation with entity modeling. In *ACL*, pages 2023–2035.

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

Paul Resnick and Hal R Varian. 1997. Recommender systems. *Communications of the ACM*, 40(3):56–58.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*, pages 379–389.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Vasu Sharma, Harsh Sharma, Ankita Bishnu, and Labhesh Patel. 2018. Cyclegen: Cyclic consistency based product review generator from attributes. In *INLG*, pages 426–430.

Yiping Song, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, Dongyan Zhao, and Rui Yan. 2018. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4382–4388.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Lex Weaver and Nigel Tao. 2001. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 538–545. Morgan Kaufmann Publishers Inc.

Jason Weston, Emily Dinan, and Alexander H. Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI, SCAI@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 87–92.

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

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. In *EMNLP*, pages 2253–2263.

Hongyu Zang and Xiaojun Wan. 2017. Towards automatic generation of product reviews from aspect-sentiment scores. In *INLG*, pages 168–177.

Rui Zhao and Kezhi Mao. 2017. Topic-aware deep compositional models for sentence classification. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 25(2):248–260.

Qingfu Zhu, Lei Cui, Weinan Zhang, Furu Wei, and Ting Liu. 2019. Retrieval-enhanced adversarial training for neural response generation. In *ACL*, pages 3763–3773.