

Generating Long and Informative Reviews with Aspect-Aware Coarse-to-Fine Decoding

Junyi Li¹, Wayne Xin Zhao^{1,2*}, Ji-Rong Wen^{1,2}, and Yang Song³

¹School of Information, Renmin University of China

²Beijing Key Laboratory of Big Data Management and Analysis Methods

³Boss Zhipin

{lijunyi, jrwen}@ruc.edu.cn

batmanfly@gmail.com songyang@kanzhun.com

Abstract

Generating long and informative review text is a challenging natural language generation task. Previous work focuses on word-level generation, neglecting the importance of topical and syntactic characteristics from natural languages. In this paper, we propose a novel review generation model by characterizing an elaborately designed aspect-aware coarse-to-fine generation process. First, we model the aspect transitions to capture the overall content flow. Then, to generate a sentence, an aspect-aware sketch will be predicted using an aspect-aware decoder. Finally, another decoder fills in the semantic slots by generating corresponding words. Our approach is able to jointly utilize aspect semantics, syntactic sketch, and context information. Extensive experiments results have demonstrated the effectiveness of the proposed model.

1 Introduction

In the past decades, online review services (*e.g.*, AMAZON and YELP) have been an important kind of information platforms where users post their feedbacks or comments about products (Kim et al., 2016). Usually, writing an informative and well-structured review will require considerable efforts by users. To assist the writing process, the task of review generation has been proposed to automatically generate review text for a user given a product and her/his rating on it (Tang et al., 2016; Zhou et al., 2017).

In the literature, various methods have been developed for review generation (Tang et al., 2016; Zhou et al., 2017; Ni et al., 2017; Wang and Zhang, 2017; Catherine and Cohen, 2018). Most of these methods adopt Recurrent Neural Networks (RNN) based methods, especially

the improved variants of Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014). They fulfill the review generation task by performing the decoding conditioned on useful context information. Usually, an informative review is likely to consist of multiple sentences, containing substantive comments from users. Hence, a major problem of existing RNN-based methods is that they have limited capacities in producing long and informative text. More recently, Generative Adversarial Net (GAN) based methods (Zang and Wan, 2017; Yu et al., 2017; Guo et al., 2018; Xu et al., 2018a) have been proposed to enhance the generation of long, diverse and novel text. However, they still focus on word-level generation, and neglect the importance of topical and syntactic characteristics from natural languages.

As found in the literature of linguistics (Pullum, 2010) and writing (Bateman and Zock, 2003), the writing process itself has involved multiple stages focusing on different levels of goals. We argue that an ideal review generation approach should follow the writing procedure of a real user and capture rich characteristics from natural language. With this motivation, we design an elaborative coarse-to-fine generation process by considering the aspect semantics and syntactic characteristics. Figure 1 presents an illustrative example for our review generation process. First, we conceive the content flow that is characterized as an aspect sequence. An aspect describes some property or attribute about a product (Zhao et al., 2010), such as *sound* and *service* in this example. To generate a sentence, we further create a sentence skeleton containing semantic slots given the aspect semantics. The semantic slots denote the placeholders for useful syntactic information (*e.g.*, Part-of-speech tags). Finally, the semantic slots are filled with the generated words. The process is repeated

*Corresponding author

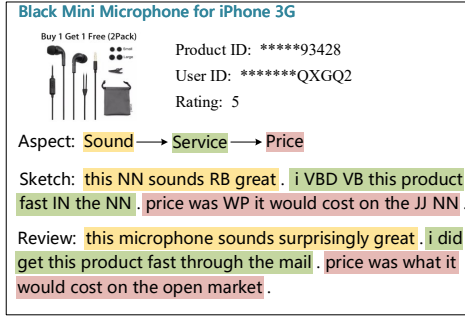


Figure 1. An illustrative example for our generation process. We select a sample review on AMAZON. The aspect labels and sketches are manually created for explaining our idea, which will be learned by our model.

until all sentences are generated.

Based on such a generation process, in this paper, we propose a novel aspect-aware coarse-to-fine decoder for generating product reviews. We first utilize unsupervised topic models to extract aspects and tag review sentences with aspect labels. We develop an attention-based RNN decoder to generate the aspect sequence conditioned on the context including users, items and ratings. By modeling the transitions of aspect semantics among sentences, we are able to capture the content flow of the whole review. Then, we generate a semantic template called *sketch* using an aspect-aware decoder, which represents the sentence skeleton. Finally, we generate the word content according to an informed decoder that considers aspect labels, sketch symbols and previously decoded words. Extensive experiments on three real-world review datasets have demonstrated the effectiveness of the proposed model.

To our knowledge, it is the first review generation model that is able to jointly utilize aspect semantics, syntactic sketch, and context information. We decompose the entire generation process into three stages. In this way, the generation of long review text becomes more controllable, since we consider a simpler sequence generation task at each stage. Furthermore, we incorporate language characteristics (*e.g.*, Part-of-Speech tags and n -grams) into the aspect-aware decoder to instruct the generation of well-structured text.

2 Related Work

In recent years, researchers have made great progress in natural language generation (NLG) (Zhang et al., 2018; Zhou et al., 2018; Fan et al., 2018). As a special NLG task, automatic re-

view generation has been proposed to assist the writing of online reviews for users. RNN-based methods have been proposed to generate the review content conditioned on useful context information (Tang et al., 2016; Zhou et al., 2017). Especially, the task of review generation is closely related to the studies in recommender systems that aim to predict the preference of a user over products. Hence, several studies propose to couple the solutions of the two lines of research work, and utilize the user-product interactions for improving the review generation (Ni et al., 2017; Wang and Zhang, 2017; Catherine and Cohen, 2018; Ni and McAuley, 2018). Although Ni and McAuley (2018) have explored aspect information to some extent, they characterize the generation process in a single stage and do not perform the coarse-to-fine decoding. Besides, the aspect transition patterns have been not modeled.

It has been found that RNN models tend to generate short, repetitive, and dull texts (Lin et al., 2018; Luo et al., 2018). For addressing this issue, Generative Adversarial Nets (GAN) based approaches have been recently proposed to generate long, diverse and novel text (Zang and Wan, 2017; Yu et al., 2017; Guo et al., 2018; Xu et al., 2018a). These methods usually utilize reinforcement learning techniques to deal with the generation of discrete symbols. However, they seldom consider the linguistic information from natural languages, which cannot fully address the difficulties of our task.

Our work is inspired by the work of using sketches as intermediate representations (Dong and Lapata, 2018; Wiseman et al., 2018; Xu et al., 2018b; Su et al., 2018). These works usually focus on sentence- or utterance-level generation tasks, in which global aspect semantics and transitions have not been considered. Our work is also related to review data mining, especially the studies on topic or aspect extraction from review data (Qiu et al., 2017; Zhao et al., 2010).

3 Problem Formulation

A review is a natural language text written by a user u on a product (or item) i with a rating score of r . Let \mathcal{V} denote the vocabulary and $y^{1:m} = \{\langle y_{j,1}, \dots, y_{j,t}, \dots, y_{j,n_j} \rangle\}_{j=1}^m$ denote a review text consisting of m sentences, where $y_{j,t} \in \mathcal{V}$ denotes the t -th word of the j -th review sentence and n_j is the length of the j -th sentence.

We assume that the review generation process is decomposed into three different stages. First, a user generates an aspect sequence representing the major content flow for a review. To generate a sentence, we predict an aspect-aware sketch conditioned on an aspect label. Finally, based on the aspect label and the sketch, we generate the word content for a sentence. The process is repeated until all the sentences are generated.

Let \mathcal{A} denote a set of A aspects in our collection. Following (Zhao et al., 2010), we assume each review sentence is associated with an aspect label, describing some property or attribute about a product or an item. We derive an aspect sequence for a review text, denoted by $a^{1:m} = \langle a_1, \dots, a_j, \dots, a_m \rangle$, where $a_j \in \mathcal{A}$ is the aspect label (or ID) of the j -th sentence. For each sentence, we assume that it is written according to some semantic sketch, which is also denoted by a symbol sequence. Let $s^{1:m} = \{\langle s_{j,1}, \dots, s_{j,t}, \dots, s_{j,n'_j} \rangle\}_{j=1}^m$, where n'_j is the length of the j -th sketch, and $s_{j,t}$ is the t -th token of the j -th sketch denoting a word, a Part-of-Speech tag, a bi-gram, etc.

Based on the above notations, we are ready to define our task. Given user u , item i and the rating score r , we aim to automatically generate a review that is able to maximize the joint probability of the aspects, sketches and words

$$\begin{aligned} & \Pr(y^{1:m}, s^{1:m}, a^{1:m} | c) \\ &= \Pr(a^{1:m} | c) \Pr(s^{1:m} | a^{1:m}, c) \Pr(y^{1:m} | a^{1:m}, s^{1:m}, c), \\ &= \prod_{j=1}^m \Pr(a_j | a_{<j}, c) \prod_{j,t} \Pr(s_{j,t} | s_{j,<t}, a_j, c) \\ & \quad \prod_{j,t} \Pr(y_{j,t} | y_{j,<t}, s_{j,t}, a_j, c), \end{aligned} \quad (1)$$

where $c = \{u, i, r\}$ denotes the set of available context information. Note that, in training, we have aspects and sketches available, and learn the model parameters by optimizing the joint probability in Eq. 1 over all the seen reviews. While, for test, the aspects and sketches are unknown. We need to first infer an aspect sequence and then predict the corresponding sketch for each sentence. Finally, we generate the review content based on the predicted aspect and sketch information.

4 The Proposed Approach

Unlike previous works generating the review in a single stage, we decompose the generation pro-

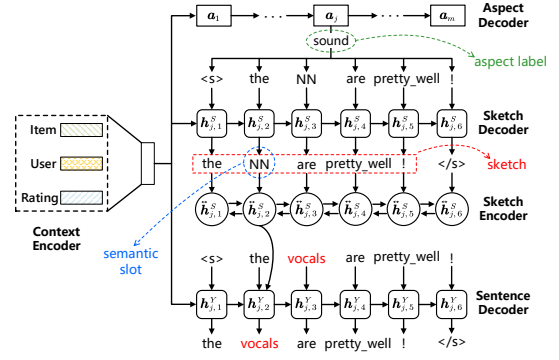


Figure 2. The overview of the proposed review generation model with the example of “the vocals are pretty well”. The predicted aspect label is *sound*, and the generated sketch is “the NN are pretty_well”.

cess into three stages, namely aspect sequence generation, aspect-aware sketch generation and sketch-based sentence generation. We present an overview illustration of the proposed model in Fig. 2. Next we describe each part in detail.

4.1 Aspect Sequence Generation

To learn the model for generating aspect sequences, we need to derive the aspect sequence for training, and then decode the aspect sequence based on the context encoder.

Aspect Extraction. Aspects provide an informative summary about the feature or attribute information about a product or an item. For example, aspects of a restaurant may include food, staff and price, etc. It is time-consuming and laborious to manually discover the aspects from texts. Here, we use an automatic unsupervised topic modeling approach to learning the aspects from the review content. Based on the Twitter-LDA model (Zhao et al., 2011), we treat a review as a document consisting of multiple sentences. Each document is associated with a distribution over the aspects. When generating a sentence, an aspect label (or ID) is first sampled according to the document’s distribution over the aspects. Then, the entire sentence is generated according to the word distribution conditioned on the aspect label. To purify the aspect words, we further incorporate a background language model to absorb background words. When topic models have been learned, we can derive a set of A aspect-specific word distributions, denoted by $\{\theta^a\}$, where θ_w^a denotes the probability of a word w from the vocabulary \mathcal{V} in aspect a .

Context Encoder. Our aspect generation module adopts an encoder-decoder architecture. We first develop the context encoder based on the information of user u , item i and rating score r . We first use a look-up layer to transform the three kinds of information into low-dimensional vectors. Let $\mathbf{v}_u \in \mathbb{R}^{d_E}$, $\mathbf{v}_i \in \mathbb{R}^{d_E}$ and $\mathbf{v}_r \in \mathbb{R}^{d_E}$ denote the embeddings for u , i and r respectively. Then, we feed the concatenated vector into a Multi-Layer Perceptron (MLP) and produce a single vectorized representation $\mathbf{v}_c \in \mathbb{R}^{d_C}$:

$$\mathbf{v}_c = \text{MLP}([\mathbf{v}_u; \mathbf{v}_i; \mathbf{v}_r]). \quad (2)$$

The embedding \mathbf{v}_c summarizes the necessary information from the three kinds of context data. It is flexible to incorporate more kinds of useful information using a similar approach.

Aspect Decoder. The decoder is built upon the GRU-based RNN network. Let $\mathbf{h}_j^A \in \mathbb{R}^{d_{HA}}$ denote a d_{HA} -dimensional hidden vector at the j -th time step, which is computed via:

$$\mathbf{h}_j^A = \text{GRU}(\mathbf{h}_{j-1}^A, \mathbf{v}_{a_{j-1}}), \quad (3)$$

where $\mathbf{v}_{a_{j-1}} \in \mathbb{R}^{d_A}$ is the embedding of the previous aspect label a_{j-1} . The hidden vector of the first time step is initialized by the encoding vector $\mathbf{h}_0^A = \mathbf{v}_c$ in Eq. 2. Then, RNNs recurrently compute hidden vectors, and predict the next aspect label (or ID) a_j . Additionally, we use an attention mechanism (Luong et al., 2015) to enhance the effect of context information. We compute the attention score of context c_k for the current time step of the decoder via:

$$w_k^{(t)} = \frac{\exp(\tanh(\mathbf{W}_1[\mathbf{h}_t^A; \mathbf{v}_{c_k}]))}{\sum_{c_{k'} \in \{u, i, r\}} \exp(\tanh(\mathbf{W}_1[\mathbf{h}_t^A; \mathbf{v}_{c_{k'}}]))}, \quad (4)$$

where \mathbf{W}_1 is the parameter matrix to learn, and the attention vector $\tilde{\mathbf{c}}_t$ is obtained by:

$$\tilde{\mathbf{c}}_t = \sum_{c_k \in \{u, i, r\}} w_k^{(t)} \mathbf{v}_{c_k} \quad (5)$$

Finally, we compute the probability of the j -th aspect label $p(a_t | a_{<j}, c)$ via:

$$\Pr(a_j | a_{<j}, c) = \text{softmax}(\mathbf{W}_4 \tilde{\mathbf{h}}_j^A + \mathbf{b}_1), \quad (6)$$

$$\tilde{\mathbf{h}}_j^A = \tanh(\mathbf{W}_2 \tilde{\mathbf{c}}_j + \mathbf{W}_3 \mathbf{h}_j^A), \quad (7)$$

where \mathbf{W}_2 , \mathbf{W}_3 , \mathbf{W}_4 and \mathbf{b}_1 are learnable parameter matrices or vector.

4.2 Aspect-Aware Sketch Generation

A sketch is a symbol sequence describing the skeleton of a sentence, where each symbol denotes a semantic symbol such as a POS tag or a bi-gram. Similar to the aspect decoder, we also use the GRU-based RNNs to implement the sketch decoder. As shown in Fig. 1, the sketches *w.r.t.* varying aspects are likely to be different. Hence, we need to consider the effect of aspect information in the generation of a sketch. Let $\mathbf{h}_{j,t}^S \in \mathbb{R}^{d_{HS}}$ denote a d_{HS} -dimensional hidden vector at time step t for the j -th sketch, which is computed via:

$$\mathbf{h}_{j,t}^S = \text{GRU}(\mathbf{h}_{j,t-1}^S, \mathbf{x}_{j,t}^S), \quad (8)$$

where $\mathbf{x}_{j,t}^S$ is further defined as

$$\mathbf{x}_{j,t} = \mathbf{v}_{s_{j,t-1}} \odot \mathbf{v}_{a_j}, \quad (9)$$

where $\mathbf{v}_{s_{j,t-1}} \in \mathbb{R}^{d_S}$ denotes the embedding for the previous sketch symbol $s_{j,t-1}$, \mathbf{v}_{a_j} denotes the embedding of the current aspect, and “ \odot ” denotes the element-wise product. In this way, the aspect information can be utilized at each time step for generating an entire sketch. We set the initial hidden vector for the j -th sketch as the last embedding of the previous sketch: $\mathbf{h}_{j,0}^S = \mathbf{h}_{j-1,n'_{j-1}}^S$. Specifically, we have $\mathbf{h}_{1,0}^S = \mathbf{v}_c$ for initialization.

Similar to Eq. 4 and 5, we can further use an attention mechanism for incorporating context information, and produce a context-enhanced sketch representation $\tilde{\mathbf{h}}_{j,t}^S$ for time step t . Finally, we compute $\Pr(s_{j,t} | s_{j,<t}, a_j, c)$ via:

$$\Pr(s_{j,t} | s_{j,<t}, a_j, c) = \text{softmax}(\mathbf{W}_5 \tilde{\mathbf{h}}_{j,t}^S + \mathbf{W}_6 \mathbf{v}_{a_j} + \mathbf{b}_2), \quad (10)$$

where we incorporate the embedding \mathbf{v}_{a_j} of the aspect a_j for enhancing the aspect semantics.

4.3 Sketch-based Review Generation

When the aspect sequence and the sketches are learned, we can generate the word content of a review. Here, we focus on the generation process of a single sentence.

Sketch Encoder. To encode the sketch information, we employ the a bi-directional GRU encoder (Schuster and Paliwal, 1997; Cho et al., 2014) to encode the sketch sequence $s_{j,1:n'_j}$ into a

list of hidden vectors $\{\overleftrightarrow{\mathbf{h}}_{j,t}^S\}_{t=1}^{n'_j}$, where $\overleftrightarrow{\mathbf{h}}_{j,t}^S$ denotes the hidden vector for the t -th position in the j -th sketch at time step t from the encoder. Different from Eq. 8, we use a bi-directional encoder

since the sketch is available at this stage, capturing the global information from the entire sketch.

Sentence Decoder. Consider the word generation at time step t . Let $\mathbf{v}_{y_{j,t-1}} \in \mathbb{R}^{d_Y}$ denotes the embedding of the previous word $y_{j,t-1}$. As input, we concatenate the current sketch representation and the embedding of the previous word

$$\mathbf{x}_{j,t}^Y = \overleftarrow{\mathbf{h}}_{j,t}^S \oplus \mathbf{v}_{y_{j,t-1}}, \quad (11)$$

where “ \oplus ” denotes the vector concatenation. Then, we compute the hidden vector $\mathbf{h}_{j,t}^Y \in \mathbb{R}^{d_{HY}}$ for the j -th sentence via:

$$\mathbf{h}_{j,t}^Y = \text{GRU}(\mathbf{h}_{j,t-1}^Y, \mathbf{x}_{j,t}^Y). \quad (12)$$

Similar to Eq. 4 and 5, we further leverage the context to obtain an enhanced state representation denoted by $\tilde{\mathbf{h}}_{j,t}^Y$ using the attention mechanism. Then we transform it into an intermediate vector with the dimensionality of the vocabulary size:

$$\mathbf{z} = \tanh(\mathbf{W}_7[\tilde{\mathbf{h}}_{j,t}^Y; \mathbf{v}_{s_{j,t}}] + \mathbf{b}_3), \quad (13)$$

where $\mathbf{v}_{s_{j,t}}$ is the embedding of the sketch symbol $s_{j,t}$. By incorporating aspect-specific word distributions, we can apply the softmax function to derive the generative probability of the t -th word

$$\Pr(y_{j,t} | y_{j,<t}, s_{j,1:n_j'}, a_j, c) = \text{softmax}(z_{y_{j,t}} + \theta_{y_{j,t}}^{a_j}), \quad (14)$$

where $\theta_{y_{j,t}}^{a_j}$ is the probability from the word distribution for aspect a_j . Here, we boost the importance of the words which have large probabilities in the corresponding topic models. In this process, the generation of words is required to match the generation of sketch symbols slot by slot. Here, we align words and sketch symbols by using the same indices for each slot for ease of understanding. However, the length of the sketch is not necessarily equal to that of the generated sentence, since a sketch symbol can correspond to a multi-term phrase. When the sketch token is a term or a phrase (e.g., bi-grams), we directly copy the original terms or phrases to the output slot(s).

4.4 Training and Inference

Integrating Eq. 6, 10 and 14 into Eq. 1, we derive the joint model for review generation. We take the log likelihood of Eq. 1 over all training reviews as the objective function. The joint objective function is difficult to be directly optimized. Hence, we

Datasets	#Users	#Items	#Reviews	#Words
AMAZON	89,672	31,829	681,004	22,570
YELP	95,617	37,112	1,063,420	31,861
RATEBEER	12,266	51,365	2,487,369	42,757

Table 1. Statistics of our datasets after preprocessing.

incrementally train the three parts, and fine-tune the shared or dependent parameters in different modules with the joint objective. For training, we directly use the real aspects and sketches for learning the model parameters. For inference, we apply our model in a pipeline way: we first infer the aspect, then predict the sketches and finally generate the words using inferred aspects and sketches. During inference, for sequence generation, we apply the beam search method with beam size 4.

In the three sequence generation modules of our model, we incorporate two special symbols to indicate the start and end of a sequence, namely START and END. Once we generate the END symbol, the generation process will be stopped. Besides, we set the maximum generation lengths for aspect sequence and sketch sequence to be 5 and 50, respectively. In the training procedure, we adopt the Adam optimizer (Kingma and Ba, 2014). In order to avoid overfitting, we adopt the dropout strategy with a rate of 0.2. More implementation details can be found in Section 5.1 (see Table 2).

5 Experiments

In this section, we first set up the experiments, and then report the results and analysis.

5.1 Experimental Setup

Datasets. We evaluate our model on three real-world review datasets, including AMAZON Electronic dataset (He and McAuley, 2016), YELP Restaurant dataset¹, and RATEBEER dataset (McAuley et al., 2012). We convert all text into lowercase, and perform tokenization using NLTK². We keep the words occurring at least ten times as vocabulary words. We discard reviews with more than 100 tokens, and remove users and products (or items) occurring fewer than five times. The reviews of each dataset are randomly split into training, validation and test sets (80%/10%/10%). The detailed statistics of the three datasets are summarized in Table 1.

¹<https://www.yelp.com/dataset>

²<https://www.nltk.org>

Modules	Settings
Aspect	$d_A = 512, d_E = 512, d_{H_A} = 512$, #GRU-layer=2, batch-size=1024, init.-learning-rate=0.00002, Adam optimizer
Sketch	$d_S = 512, d_{H_S} = 512$, #GRU-layer=2, batch-size=64, init.-learning-rate=0.0002, learning-rate-decay-factor=0.8, learning-rate-decay-epoch=2, Adam optimizer
Review	$d_Y = 512, d_{H_Y} = 512$, #GRU-layer=2, batch-size=64, init.-learning-rate=0.0002, learning-rate-decay-factor=0.8, learning-rate-decay-epoch=2, Adam optimizer

Table 2. Parameter settings of the three modules in our model.

Aspect and Sketch Extraction. After the preprocessing, we use the Twitter-LDA model in (Zhao et al., 2011) for automatically learning the aspects and aspect keywords. The numbers of aspects are set to 10, 5, and 5 for the three datasets, respectively. The aspect numbers are selected using the perplexity score on validation set. By inspecting into the top aspect words, we find the learned aspects are very coherent and meaningful. For convenience, we ask a human labeler to annotate each learned aspect from topic models with an *aspect label*. Note that aspect labels are only for ease of presentation, and will not be used in our model. With topic models, we further tag each sentence with the aspect label which gives the maximum posterior probability conditioned on the words. To derive the sketches, we first extract the most popular 200 bi-grams and tri-grams by frequency. We replace their occurrences with n -gram IDs. Furthermore, we keep the words ranked in top 50 positions of an aspect, and replace the occurrences of the rest words with their Part-of-Speech tags. We also keep the top 50 frequent words in the entire text collection, such as background words “I” and “am”. In this way, for each review, we obtain a sequence of aspect labels; for each sentence in the review, we obtain a sequence of sketch symbols. Aspect sequences and sketch sequences are only available during the training process.

Baseline Models. We compare our model against a number of baseline models:

- gC2S (Tang et al., 2016): It adopts an encoder-decoder architecture to generate review texts conditioned on context information through a gating mechanism.
- Attr2Seq (Zhou et al., 2017): It adopts an attention-enhanced attribute-to-sequence architec-

ture to generate reviews with input attributes.

- TransNets (Catherine and Cohen, 2018): It applies a student-teacher like architecture for review generation by representing the reviews of a user and an item into a text-related representation, which is regularized to be similar to the actual review’s latent representation at training time.

- ExpansionNet (Ni and McAuley, 2018): It uses an encoder-decoder framework to generate personalized reviews by incorporating short phrases (e.g., review summaries, product titles) provided as input and introducing aspect-level information (e.g., aspect words).

- SeqGAN (Yu et al., 2017): It regards the generative model as a stochastic parameterized policy and uses Monte Carlo search to approximate the state-action value. The discriminator is a binary classifier to evaluate the sequence and guide the learning of the generative model.

- LeakGAN (Guo et al., 2018): The generator is built upon a hierarchical reinforcement learning architecture, which consists of a high-level module and a low-level module, and the discriminator is a CNN-based feature extractor. The advantage is that this model can generate high-quality long text by introducing the leaked mechanism.

Among these baselines, gC2S, Attr2Seq and TransNets are context-aware generation models in different implementation approaches, ExpansionNet introduces external information such as aspect words, and SeqGAN and LeakGAN are GAN based text generation models. Original SeqGAN and LeakGAN are designed for general sequence generation without considering context information (e.g., user, item, rating). The learned aspect keywords are provided as input for both ExpansionNet and our model. All the methods have several parameters to tune. We employ validation set to optimize the parameters in each method. To reproduce the results of our model, we report the parameter setting used throughout the experiments in Table 2. Our code is available at <https://github.com/turboLJY/Coarse-to-Fine-Review-Generation>.

Evaluation Metrics. To evaluate the performance of different methods on automatic review generation, we adopt six evaluation metrics, including Perplexity, BLEU-1/BLEU-4, ROUGE-1/ROUGE-2/ROUGE-L. Perplexity³ is the standard measure for evaluating language models;

³<https://en.wikipedia.org/wiki/Perplexity>

Datasets	Models	Perplexity	BLEU-1(%)	BLEU-4(%)	ROUGE-1	ROUGE-2	ROUGE-L
AMAZON	gC2S	38.67	24.14	0.85	0.262	0.046	0.212
	Attr2Seq	34.67	24.28	0.88	0.263	0.043	0.214
	TransNets	34.21	21.61	0.60	0.227	0.026	0.199
	ExpansionNet	31.50	26.56	0.95	0.290	0.052	0.262
	SeqGAN	28.50	25.18	0.84	0.265	0.043	0.220
	LeakGAN	27.66	25.66	0.92	0.267	0.050	0.236
	Our model	26.55	28.22	1.04	0.315	0.066	0.280
YELP	gC2S	35.52	24.39	0.87	0.243	0.046	0.188
	Attr2Seq	33.12	24.71	0.89	0.245	0.047	0.191
	TransNets	34.81	21.41	0.35	0.202	0.026	0.156
	ExpansionNet	29.53	27.46	1.06	0.276	0.061	0.216
	SeqGAN	26.84	24.83	0.99	0.253	0.054	0.192
	LeakGAN	25.53	25.96	1.03	0.271	0.056	0.208
	Our model	23.96	29.43	1.13	0.284	0.070	0.235
RATEBEER	gC2S	17.81	32.13	5.55	0.379	0.140	0.331
	Attr2Seq	16.84	32.21	5.80	0.380	0.142	0.331
	TransNets	19.08	29.74	3.61	0.347	0.114	0.302
	ExpansionNet	17.07	34.53	6.83	0.400	0.156	0.376
	SeqGAN	14.30	32.41	5.62	0.369	0.146	0.337
	LeakGAN	13.74	33.76	6.03	0.378	0.142	0.355
	Our model	13.07	36.11	7.04	0.422	0.164	0.393

Table 3. Performance comparisons of different methods for automatic review generation using three datasets.

BLEU (Papineni et al., 2002) measures the ratios of the co-occurrences of n -grams between the generated and real reviews; ROUGE (Lin, 2004) measures the review quality by counting the overlapping n -grams between the generated and real reviews.

5.2 Results and Analysis

In this subsection, we construct a series of experiments on the effectiveness of the proposed model for the review generation task.

Main Results. Table 3 presents the performance of different methods on automatic review generation. We can make the following observations. First, among the three context-based baselines, gC2S and Attr2Seq perform better than TransNets. The two models have similar network architectures, which are simpler than TransNets. We find they are easier to obtain a stable performance on large datasets. Second, GAN-based methods work better than the above baselines, especially LeakGAN. LeakGAN is specially designed for generating long text, and we adapt it to our task by incorporating context information. Third, ExpansionNet performs best among all the baseline models. A major reason is that it incorporates external knowledge such as review summaries, product titles and aspect keywords. Finally, our model outperforms all the baselines with a large margin. These baseline methods perform the generation in

Models	BLEU-1(%)	ROUGE-1
Our model	28.22	0.315
w/o aspect	27.85	0.296
w/o sketch	25.95	0.273

Table 4. Ablation analysis on AMAZON dataset.

a single stage. As a comparison, we use a multi-stage process to gradually generate long and informative reviews in a coarse-to-fine way. Our model is able to better utilize aspect semantics and syntactic sketch, which is the key of the performance improvement over baselines. Overall, the three datasets show the similar findings. In what follows, we will report the results on AMAZON data due to space limit. We select the best two baselines *ExpansionNet* and *LeakGAN* as reference methods.

Ablation Analysis. The major novelty of our model is that it incorporates two specific modules to generate aspects and sketches respectively. To examine the contribution of the two modules, we compare our model with its two variants by removing either of the two modules. We present the BLEU-1 and ROUGE-1 results of our model and its two variants in Table 4. As we can see, both components are useful to improve the final performance, and the sketch generation module seems more important in our task. In our model, the aspect generation module is used to cover aspect semantics and generate informative review;

the sketch generation module is able to utilize syntactic templates to improve the generation fluency, especially for long sentences. Current experiments evaluate the usefulness of the two modules based on the overall generation quality. Next, we verify their functions using two specific experiments, namely aspect coverage and fluency evaluation.

Aspect Coverage Evaluation. A generated review is informative if it can effectively capture the semantic information of the real review. Following (Ni and McAuley, 2018), we examine the aspect coverage of different models. Recall that we have used topic models to tag each sentence with an aspect label (or ID). We analyze the average number of aspects in real and generated reviews, and compute on average how many aspects in real reviews are covered in generated reviews. We consider a review as covering an aspect if any of the top 50 words of an aspect exists in the review⁴. In Table 5, we first see an interesting observation that LeakGAN is able to generate more aspects but yield fewer real aspects. As a comparison, ExpansionNet and our model perform better than LeakGAN by covering more real aspects, since the two models use the aspect information to instruct the review generation. Our model is better than ExpansionNet by characterizing the aspect transition sequences. These results indicate the usefulness of the aspect generation module in capturing more semantic information related to a review.

Fluency Evaluation. We continue to evaluate the usefulness of the sketch generation module in improving the fluency of the generated text. Following (Xu et al., 2018a), we construct the fluency evaluation to examine how likely the generated text is produced by human. We randomly choose 200 samples from test set. A sample contains the input contexts (*i.e.*, user, item, rating), and the texts generated by different models. It is difficult to develop automatic evaluation methods for accurate fluency evaluation. Here, we invite two human annotators (excluding the authors of this paper) who have good knowledge in the domain of electronic reviews to assign scores to the generated reviews. They are required to assign a score to a generated (or real) review according to a 5-point Likert scale⁵ on fluency. In the 5-point Lik-

Models	# aspects (real)	# aspects (generated)	# covered aspects
ExpansionNet	2.41	2.02	0.885
LeakGAN	2.41	2.18	0.630
Our model	2.41	2.03	1.076

Table 5. Aspect coverage evaluation on AMAZON dataset.

Measures	Gold	ExpansionNet	LeakGAN	Our
Fluency	4.01	3.29	3.26	3.54
Kappa	0.80	0.72	0.76	0.74

Table 6. Fluency evaluation on AMAZON dataset.

ert scale, 5-point means “very satisfying”, while 1-point means “very terrible”. We further average the two annotated scores over the 200 inputs. The results are shown in Table 6. We can see that our model achieves the highest fluency score among the automatic methods. By using sketches, our model is able to leverage the learned syntactic patterns from available reviews. The Cohen’s kappa coefficients are above 0.7, indicating a high correlation and agreement between the two human annotators.

5.3 Qualitative Analysis

In this part, we perform the qualitative analysis on the quality of the generated reviews. We present three sample reviews generated by our model in Table 7. As we can see, our model has covered most of the major aspects (with many overlapping aspect keywords) of the real reviews. Although some generated sentences do not follow the exact syntactic structures of real reviews, they are very readable to users. Our model is able to generate aspect-aware sketches, which are very helpful to instruct the generation of the word content. With the aspect and sketch generation modules, our model is able to produce informative reviews consisting of multiple well-structured sentences. Another interesting observation is that the polarities of the generated text also correspond to their real rating scores, since the rating score has been modeled in the context encoder.

6 Conclusion

This paper presented a novel review generation model using an aspect-aware coarse-to-fine generation process. Unlike previous methods, our model decomposed the generation process into three stages focusing on different goals. We constructed extensive experiments on three real-world review datasets. The results have demonstrated

⁴For accuracy, we manually remove the irrelevant words (about 5%~10%) from the top 50 words in each aspect.

⁵https://en.wikipedia.org/wiki/Likert_scale

Gold Standard	Generated Sketch	Generated Review
the shipping was quick and easy ^{service} very good product at a reasonable price ^{price} 5mm male to 2 rca stereo audio cable ^{cable} sound highly recommend this product to anyone ^{overall}	this cable worked perfectly for my NNS ^{sound} the price was very JJ and i would purchase NN from this NN ^{price} it VBD on_time and in good NN ^{service} i would recommend it ^{overall}	this cable worked perfectly for my needs ^{sound} the price was very reasonable and i would purchase another from this vendor ^{price} it arrived on time and in good condition ^{service} i would recommend it ^{overall}
oxtail was good other than the flavors were very bland ^{food} place is small so if the tables are full be prepared to wait ^{place} pay too much for what you get ^{price} i will not be back to this location ^{overall}	i had the NN NN and it was very JJ ^{food} the staff was JJ but service was a little JJ ^{service} i had a bad_experience at this NN ^{place} i VBP not JJ if i will be back RB ^{overall}	i had the falafel wrap and it was very bland ^{food} the staff was friendly but service was a little slow ^{service} i had a bad_experience at this place ^{place} i am not sure if i will be back again ^{overall}
the aroma is insanely sour from bad hops ^{aroma} dark clear ruby red beat sugar flavor and strong alcohol in aftertaste ^{flavor} golden body with a small white head ^{body} dont waste your money on this ^{overall}	VBZ an amber_body with a JJ NN head ^{body} the flavor is very JJ with notes of NN ^{flavor} this beer has the JJS aroma of canned_corn i have ever VBN ^{aroma}	pours an amber body with a white finger head ^{body} the flavor is very horrible with notes of alcohol ^{flavor} this beer has the worst aroma of canned corn i have ever smelled ^{aroma}

Table 7. Samples of the generated reviews by our model. The three reviews with rating scores of 5 (positive), 3 (neutral), and 1 (negative) are from AMAZON, YELP and RATEBEER datasets, respectively. For privacy, we omit the UIDs and PIDs. For ease of reading, colored aspect labels are manually created corresponding to the predicted aspect IDs by our model. We have underlined important overlapping terms between real and generated reviews.

the effectiveness of our model in terms of overall generation quality, aspect coverage, and fluency. As future work, we will consider integrating more kinds of syntactic features from linguistic analysis such as dependency parsing.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61872369 and 61832017, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China under Grant No. 18XNLG22 and 19XNQ047.

References

- John Bateman and Michael Zock. 2003. Natural language generation. In *The Oxford Handbook of Computational Linguistics 2nd edition*.
- Rose Catherine and William Cohen. 2018. Transnets for review generation.
- Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734.
- Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 731–742.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898.
- Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5141–5148.
- Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Bona Kim, Seongseop Kim, and Cindy Y Heo. 2016. Analysis of satisfiers and dissatisfiers in online hotel reviews on social media. *International Journal of Contemporary Hospitality Management*, 28(9):1915–1936.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.
- Junyang Lin, Xu Sun, Shuming Ma, and Qi Su. 2018. Global encoding for abstractive summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 163–169.

- Liangchen Luo, Jingjing Xu, Junyang Lin, Qi Zeng, and Xu Sun. 2018. An auto-encoder matching model for learning utterance-level semantic dependency in dialogue generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 702–707.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421.
- Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 1020–1025.
- Jianmo Ni, Zachary C. Lipton, Sharad Vikram, and Julian McAuley. 2017. Estimating reactions and recommending products with generative models of reviews. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 783–791.
- 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 706–711.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318.
- Geoffrey K Pullum. 2010. The land of the free and the elements of style. *English Today*, 26(2):34–44.
- Minghui Qiu, Yinfei Yang, Cen Chen, and Forrest Sheng Bao. 2017. Aspect extraction from product reviews using category hierarchy information. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 675–680.
- Mike Schuster and Kuldeep K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45(11):2673–2681.
- Shang-Yu Su, Kai-Ling Lo, Yi Ting Yeh, and Yun-Nung Chen. 2018. Natural language generation by hierarchical decoding with linguistic patterns. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 61–66.
- Jian Tang, Yifan Yang, Samuel Carton, Ming Zhang, and Qiaozhu Mei. 2016. Context-aware natural language generation with recurrent neural networks. *CoRR*, abs/1611.09900.
- Zhongqing Wang and Yue Zhang. 2017. Opinion recommendation using A neural model. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1626–1637.
- Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3174–3187.
- Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018a. Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3940–3949.
- Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018b. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4306–4315.
- Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 2852–2858.
- Hongyu Zang and Xiaojun Wan. 2017. Towards automatic generation of product reviews from aspect-sentiment scores. In *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, pages 168–177.
- Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2018. Learning to control the specificity in neural response generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1108–1117.
- Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011*,

Dublin, Ireland, April 18-21, 2011. Proceedings, pages 338–349.

Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. 2010. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010*, pages 56–65.

Ming Zhou, Mirella Lapata, Furu Wei, Li Dong, Shaohan Huang, 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, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 623–632.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 654–663.