Personalized Review Generation by Expanding Phrases and Attending on Aspect-Aware Representations

Jianmo Ni
University of California San Diego
jin018@ucsd.edu

Julian McAuley
University of California San Diego
jmcauley@ucsd.edu

Abstract

In this paper, we focus on the problem of building assistive systems that can help users to write reviews. We cast this problem using an encoder-decoder framework that generates personalized reviews by expanding short phrases (e.g. review summaries, product titles) provided as input to the system. We incorporate aspect-level information via an aspect encoder that learns 'aspect-aware' user and item representations. An attention fusion layer is applied to control generation by attending on the outputs of multiple encoders. Experimental results show that our model is capable of generating coherent and diverse reviews that expand the contents of input phrases. In addition, the learned aspect-aware representations discover those aspects that users are more inclined to discuss and bias the generated text toward their personalized aspect preferences.

1 Introduction

Contextual, or 'data-to-text' natural language generation is one of the core tasks in natural language processing and has a considerable impact on various fields (Gatt and Krahmer, 2017). Within the field of recommender systems, a promising application is to estimate (or generate) personalized reviews that a user would write about a product, i.e., to discover their nuanced opinions about each of its individual aspects. A successful model could work (for instance) as (a) a highly-nuanced recommender system that tells users their likely reaction to a product in the form of text fragments; (b) a writing tool that helps users 'brainstorm' the review-writing process; or (c) a querying system that facilitates personalized natural language queries (i.e., to find items about which a user would be most likely to write a particular phrase). Some recent works have explored the review generation task and shown success in generating cohesive reviews (Dong et al., 2017; Ni et al., 2017; Zang and Wan, 2017). Most of these works treat the user and item identity as input; we seek a system with more nuance and more precision by allowing users to 'guide' the model via short phrases, or auxiliary data such as item specifications. For example, a review writing assistant might allow users to write short phrases and expand these key points into a plausible review.

Review text has been widely studied in traditional tasks such as aspect extraction (Mukherjee and Liu, 2012; He et al., 2017), extraction of sentiment lexicons (Zhang et al., 2014), and aspect-aware sentiment analysis (Wang et al., 2016; McAuley et al., 2012). These works are related to review generation since they can provide prior knowledge to supervise the generative process. We are interested in exploring how such knowledge (e.g. extracted aspects) can be used in the review generation task.

In this paper, we focus on designing a review generation model that is able to leverage both user and item information as well as auxiliary, textual input and aspect-aware knowledge. Specifically, we study the task of expanding short phrases into complete, coherent reviews that accurately reflect the opinions and knowledge learned from those phrases.

These short phrases could include snippets provided by the user, or manifest aspects about the items themselves (e.g. brand words, technical specifications, etc.). We propose an encoder-decoder framework that takes into consideration three encoders (a sequence encoder, an attribute encoder, and an aspect encoder), and one decoder. The sequence encoder uses a gated recurrent unit

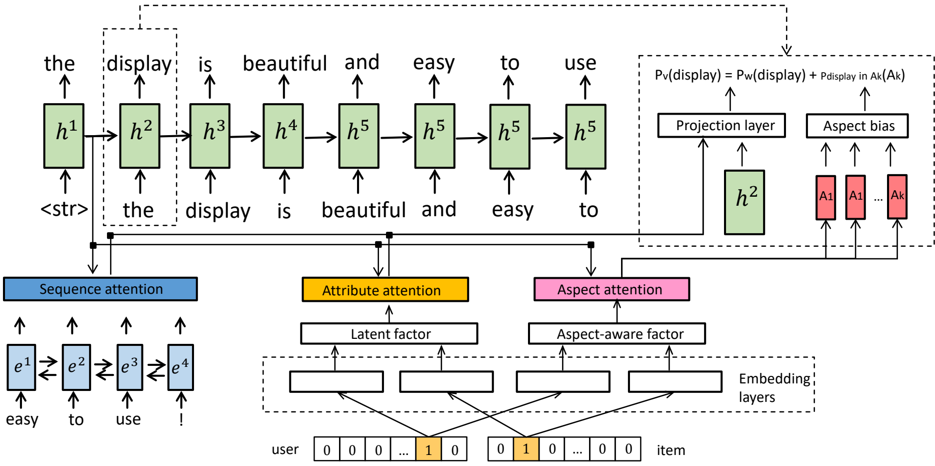

Figure 1: General structure of ExpansionNet.

(GRU) network to encode text information; the attribute encoder learns a latent representation of user and item identity; finally, the aspect encoder finds an aspect-aware representation of users and items, which reflects user-aspect preferences and item-aspect relationships. The aspect-aware representation is helpful to discover what each user is likely to discuss about each item. Finally, the output of these encoders is passed to the sequence decoder with an attention fusion layer. The decoder attends on the encoded information and biases the model to generate words that are consistent with the input phrases and words belonging to the most relevant aspects.

2 Related Work

Review generation belongs to a large body of work on data-to-text natural language generation (Gatt and Krahmer, 2017), which has applications including summarization (See et al., 2017), image captioning (Vinyals et al., 2015), and dialogue response generation (Xing et al., 2017; Li et al., 2016; Ghosh et al., 2017), among others. Among these, review generation is characterized by the need to generate long sequences and estimate high-order interactions between users and items.

Several approaches have been recently proposed to tackle these problems. Dong et al. (2017) proposed an attribute-to-sequence (Attr2Seq) method to encode user and item identities as well as rating information with a multi-layer perceptron and a decoder then generates reviews conditioned on this information. They also used an attention mechanism to strengthen the alignment between output and input attributes. Ni et al. (2017) trained a collaborative-filtering generative concatenative network to jointly learn the tasks of review generation and item recommendation. Zang and Wan (2017) proposed a hierarchical structure to generate long reviews; they assume each sentence is associated with an aspect score, and learn the attention between aspect scores and sentences during training. Our approach differs from these mainly in our goal of incorporating auxiliary textual information (short phrases, product specifications, etc.) into the generative process, which facilitates the generation of higher-fidelity reviews.

Another line of work related to review generation is aspect extraction and opinion mining (Park et al., 2015; Qiu et al., 2017; He et al., 2017; Chen et al., 2014). In this paper, we argue that the extra aspect (opinion) information extracted using these previous works can effectively improve the quality of generated reviews. We propose a simple but effective way to combine aspect information into the generative model.

3 Approach

We describe the review generation task as follows. Given a user u, item i, several short phrases $\{d_1, d_2, ..., d_{\mathrm{M}}\}$, and a group of extracted aspects $\{A_1, A_2, ..., A_k\}$, our goal is to generate a review $(w_1, w_2, ..., w_{\mathrm{T}})$ that maximizes the probability $P(w_{1:\mathrm{T}}|u, i, d_{1:\mathrm{M}})$. To solve this task, we propose a method called *ExpansionNet* which contains two parts: 1) three encoders to leverage the input phrases and aspect information; and 2) a decoder with an attention fusion layer to generate sequences and align the generation with the input sources. The model structure is shown in Figure 1.

3.1 Sequence encoder, attribute encoder and aspect encoder

Our sequence encoder is a two-layer bi-directional GRU, as is commonly used in sequence-to-sequence (Seq2Seq) models (Cho et al., 2014). Input phrases first pass a word embedding layer, then go through the GRU one-by-one and finally yield a sequence of hidden states $\{\mathbf{e}_1, \mathbf{e}_2..., \mathbf{e}_{\mathrm{L}}\}$. In the case of multiple phrases, these share the same sequence encoder and have different lengths L. To simplify notation, we only consider one input phrase in this section.

The attribute encoder and aspect encoder both consist of two embedding layers and a projection layer. For the attribute encoder, we define two general embedding layers $\mathrm{E_u} \in \mathbb{R}^{|\mathcal{U}| \times m}$ and $\mathrm{E_i} \in \mathbb{R}^{|\mathcal{I}| \times m}$ to obtain the attribute latent factors γ_u and γ_i; for the aspect encoder, we use two aspect-aware embedding layers $\mathrm{E_u'} \in \mathbb{R}^{|\mathcal{U}| \times k}$ and $\mathrm{E_i'} \in \mathbb{R}^{|\mathcal{I}| \times k}$ to obtain aspect-aware latent factors β_u and β_i. Here $|\mathcal{U}|$, $|\mathcal{I}|$, m and k are the number of users, number of items, the dimension of attributes, and the number of aspects, respectively. After the embedding layers, the attribute and aspect-aware latent factors are concatenated and fed into a projection layer with tanh activation. The outputs are calculated as:

$$\gamma_u = \mathrm{E_u}(u), \gamma_i = \mathrm{E_i}(i) \tag{1}$$

$$\beta_u = \mathrm{E_u'}(u), \beta_i = \mathrm{E_i'}(i) \tag{2}$$

$$\mathbf{u} = \tanh(W_{\mathbf{u}}[\gamma_u; \gamma_i] + \mathbf{b_u}) \tag{3}$$

$$\mathbf{v} = \tanh(W_{\mathbf{v}}[\beta_u; \beta_i] + \mathbf{b_v}) \tag{4}$$

where $W_{\mathbf{u}} \in \mathbb{R}^{n \times 2m}$, $\mathbf{b_u} \in \mathbb{R}^n$, $W_{\mathbf{v}} \in \mathbb{R}^{n \times 2k}$, $\mathbf{b_v} \in \mathbb{R}^n$ are learnable parameters and n is the dimensionality of the hidden units in the decoder.

3.2 Decoder with attention fusion layer

The decoder is a two-layer GRU that predicts the target words given the start token. The hidden state of the decoder is initialized using the sum of the three encoders' outputs. The hidden state at time-step t is updated via the GRU unit based on the previous hidden state and the input word. Specifically:

$$\mathbf{h}_0 = \mathbf{e}_{\mathrm{L}} + \mathbf{u} + \mathbf{v} \tag{5}$$

$$\mathbf{h}_t = \mathbf{GRU}(w_t, \mathbf{h}_{t-1}), \tag{6}$$

where $\mathbf{h}_0 \in \mathbb{R}^n$ is the decoder's initial hidden state and $\mathbf{h}_t \in \mathbb{R}^n$ is the hidden state at time-step t.

To fully exploit the encoder-side information, we apply an attention fusion layer to summarize the output of each encoder and jointly determine the final word distribution. For the sequence encoder, the attention vector is defined as in many other applications (Bahdanau et al., 2014; Luong et al., 2015):

$$\mathbf{a}_t^1 = \sum_{j=1}^{\mathrm{L}} \alpha_{tj}^1 \mathbf{e}_j \tag{7}$$

$$\alpha_{tj}^1 = \exp(\tanh(\mathbf{v}_\alpha^{1\top}(W_\alpha^1[\mathbf{e}_j; \mathbf{h}_t] + \mathbf{b}_\alpha^1)))/Z, \tag{8}$$

where $\mathbf{a}_t^1 \in \mathbb{R}^n$ is the attention vector on the sequence encoder at time-step t, α_{tj}^1 is the attention score over the encoder hidden state $\mathbf{e}_j$ and decoder hidden state $\mathbf{h}_t$, and Z is a normalization term.

For the attribute encoder, the attention vector is calculated as:

$$\mathbf{a}_t^2 = \sum_{j \in u,i} \alpha_{tj}^2 \gamma_j \tag{9}$$

$$\alpha_{tj}^2 = \exp(\tanh(\mathbf{v}_\alpha^{2\top}(W_\alpha^2[\gamma_j; \mathbf{h}_t] + \mathbf{b}_\alpha^2)))/Z, \tag{10}$$

where $\mathbf{a}_t^2 \in \mathbb{R}^n$ is the attention vector on the attribute encoder, and α_{tj}^2 is the attention score between the attribute latent factor γ_j and decoder hidden state $\mathbf{h}_t$.

Inspired by the copy mechanism (Gu et al., 2016; See et al., 2017), we design an attention vector that estimates the probability that each aspect will be discussed in the next time-step:

$$s_{ui} = W_s[\beta_u; \beta_i] + \mathbf{b}_s \tag{11}$$

$$\mathbf{a}_t^3 = \tanh(W_\alpha^3[s_{ui}; e_t; \mathbf{h}_t] + \mathbf{b}_\alpha^3), \tag{12}$$

where $s_{ui} \in \mathbb{R}^k$ is the aspect importance considering the interaction between u and i, e_t is the decoder input after embedding layer at time-step t, and $\mathbf{a}_t^3 \in \mathbb{R}^k$ is a probability vector to bias each aspect at time-step t. Finally, the first two attention vectors are concatenated with the decoder hidden state at time-step t and projected to obtain the output word distribution P_v. The attention scores from the aspect encoder are then directly added to the aspect words in the final word distribution. The output probability for word w at time-step t is given by:

$$P_v(w_t) = \tanh(W[\mathbf{h}_t; \mathbf{a}_t^1; \mathbf{a}_t^2] + \mathbf{b}) \tag{13}$$

$$P(w_t) = P_v(w_t) + \mathbf{a}_t^3[k] \cdot \mathbb{1}_{w_t \in A_k}, \tag{14}$$

Table 1: Parameter settings used in our experiments.

Word dimension	Attribute dimension	Aspect dimension	GRU hidden size	Batch Size	Learning Rate	Optimizer
512	64	15	512	25	0.0002	Adam

where w_t is the target word at time-step t, $\mathbf{a}_t^3[k]$ is the probability that aspect k will be discussed at time-step t, A_k represents all words belonging to aspect k and $\mathbb{1}_{w_t \in A_k}$ is a binary variable indicating whether w_t belongs to aspect k.

During inference, we use greedy decoding by choosing the word with maximum probability, denoted as $y_t = \text{argmax}_{w_t}\text{softmax}(P(w_t))$. Decoding finishes when an end token is encountered.

4 Experiments

We consider a real world dataset from *Amazon Electronics* (McAuley et al., 2015) to evaluate our model. We convert all text into lowercase, add start and end tokens to each review, and perform tokenization using NLTK.[1] We discard reviews with length greater than 100 tokens and consider a vocabulary of 30,000 tokens. After preprocessing, the dataset contains 182,850 users, 59,043 items, and 992,172 reviews (sparsity 99.993%), which is much sparser than the datasets used in previous works (Dong et al., 2017; Ni et al., 2017). On average, each review contains 49.32 tokens as well as a short-text summary of 4.52 tokens. In our experiments, the basic ExpansionNet uses these summaries as input phrases. We split the dataset into training (80%), validation (10%) and test sets (10%). All results are reported on the test set.

4.1 Aspect Extraction

We use the method[2] in (He et al., 2017) to extract 15 aspects and consider the top 100 words from each aspect. Table 2 shows 10 inferred aspects and representative words (inferred aspects are manually labeled). ExpansionNet calculates an attention score based on the user and item aspect-aware representation, then determines how much these representative words are biased in the output word distribution.

Table 2: List of representative words for inferred aspects on Amazon Electronics dataset.

Aspects	Representative Words
Service	vendor seller supplier reply refund delivery shipping exchange contacting promptly
Price	price value overall dependable reliable affordable practical budget inexpensive bargain
Screen	screen touchscreen browse display scrolling surfing navigate icon menu surfing text blur reflection
Case	case cover briefcase portfolio padded protective rubberized padding leather skin
Drive	drive disk copying copied fat32 terabyte ntfs data hdd cache
Sound	sound vocal loudness booming bass treble tinny speaker isolation sennheisers
Vision	glossy shiny transparent polish reflective faded lcd shield glass painted
Laptop	lenovo inspiron ibm gateway pentium alienware xps pavilion thinkpad elite
Time	cycle time week day month hour suddenly repeated overnight continuously
Stableness	unscrew securing mounting drill centered tightening screwed attach tighten loosen

4.2 Experiment Details

We use PyTorch[3] to implement our model.[4] Parameter settings are shown in Table 1. For the attribute encoder and aspect encoder, we set the dimensionality to 64 and 15 respectively. For both the sequence encoder and decoder, we use a 2-layer GRU with hidden size 512. We also add dropout layers before and after the GRUs. The dropout rate is set to 0.1. During training, the input sequences of the same source (e.g. review, summary) inside each batch are padded to the same length.

4.3 Performance Evaluation

We evaluate the model on six automatic metrics (Table 3): Perplexity, BLEU-1/BLEU-4, ROUGE-L and Distinct-1/2 (percentage of distinct unigrams and bi-grams) (Li et al., 2016). We compare

[1] https://www.nltk.org/

[2] https://github.com/ruidan/Unsupervised-Aspect-Extraction

[3] http://pytorch.org/docs/master/index.html

[4] https://github.com/nijianmo/textExpansion

Table 3: Results on automatic metrics

Model	PPL	BLEU-1(%)	BLEU-4(%)	ROUGE-L	Distinct-1(%)	Distinct-2(%)
Rand	/	20.24	0.45	0.390	1.311	13.681
GRU-LM	35.35	30.79	1.20	/	/	/
Att2Seq	34.21	26.16	1.23	0.403	0.014	0.051
+aspect	34.26	26.87	1.51	0.397	0.018	0.069
ExpansionNet	34.18	26.05	2.21	0.404	0.096	0.789
+title	**30.7**	27.90	2.50	**0.415**	0.099	0.911
+attribute & aspect	31.7	**30.33**	**2.63**	0.408	**0.133**	**1.134**

User/Item Review summary Item title	user *A3G831BTCLWGVQ* and item *B007M50PTM* "easy to use and nice standard apps" "samsung galaxy tab 2 (10.1-Inch, wi-fi) 2012 model"
Real review	"the display is beautiful and the tablet is very easy to use. it comes with some really nice standard apps."
AttrsSeq	"i bought this for my wife 's new ipad air . it fits perfectly and looks great . the only thing i do n't like is that the cover is a little too small for the ipad air . "
ExpansionNet	"i love this tablet . it is fast and easy to use . i have no complaints . i would recommend this tablet to anyone ."
+title	"i love this tablet . it is fast and easy to use . i have a galaxy tab 2 and i love it ."
+attribute & aspect	"i love this tablet . it is easy to use and the screen is very responsive . i love the fact that it has a micro sd slot . i have not tried the tablet app yet but i do n't have any problems with it . i am very happy with this tablet ."

Figure 2: Examples of a real review and reviews generated by different models given a user, item, review summary, and item title. Highlights added for emphasis.

against three baselines: Rand (randomly choose a review from the training set), GRU-LM (the GRU decoder works alone as a language model) and a state-of-the-art model Attr2Seq that only considers user and item attributes (Dong et al., 2017). ExpansionNet (with summary, item title, attribute and aspect as input) achieves significant improvements over Attr2Seq on all metrics. As we add more input information, the model continues to obtain better results, except for the ROUGE-L metric. This proves that our model can effectively learn from short input phrases and aspect information and improve the correctness and diversity of generated results.

Figure 2 presents a sample generation result. ExpansionNet captures fine-grained item information (e.g. that the item is a tablet), which Attr2Seq fails to recognize. Moreover, given a phrase like "easy to use" in the summary, ExpansionNet generates reviews containing the same text. This demonstrates the possibility of using our model in an assistive review generation scenario. Finally, given extra aspect information, the model successfully estimates that the screen would be an important aspect (i.e., for the current user and item); it generates phrases such as "screen is very responsive" about the aspect "screen" which is also covered in the real (ground-truth) review ("display is beautiful").

Table 4: Aspect coverage analysis

	# aspects (real)	# aspects (generated)	# covered aspects
Attr2Seq	2.875	**2.744**	0.686
ExpansionNet	2.875	1.804	0.807
+title	2.875	1.721	0.894
+attribute&aspect	2.875	1.834	**0.931**

We are also interested in seeing how the aspect-aware representation can find related aspects and bias the generation to discuss more about those aspects. We analyze the average number of aspects in real and generated reviews and show 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 aspect's representative words exists in the review. As shown in Table 4, Attr2Seq tends to cover more aspects in generation, many of which are not discussed in real reviews. On the other hand, ExpansionNet better captures the distribution of aspects that are discussed in real reviews.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *ACL*.

Kyunghyun Cho, Bart van Merrienboer, aglar Gülehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

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

Albert Gatt and Emiel Krahmer. 2017. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *JAIR*.

Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2017. Affect-lm: A neural language model for customizable affective text generation. In *ACL*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *ACL*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. 2016. A persona-based neural conversation model. In *ACL*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *ICDM*.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*.

Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *ACL*.

Jianmo Ni, Zachary C. Lipton, Sharad Vikram, and Julian J. McAuley. 2017. Estimating reactions and recommending products with generative models of reviews. In *International Joint Conference on Natural Language Processing*.

Dae Hoon Park, Hyun Duk Kim, ChengXiang Zhai, and Lifan Guo. 2015. Retrieval of relevant opinion sentences for new products. In *SIGIR*.

Minghui Qiu, Yinfei Yang, Cen Chen, and Forrest Sheng Bao. 2017. Aspect extraction from product reviews using category hierarchy information. In *EACL*.

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

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *EMNLP*.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *AAAI*.

Hongyu Zang and Xiaojun Wan. 2017. Towards automatic generation of product reviews from aspect-sentiment scores. In *International Conference on Natural Language Generation*.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*.