ADORE: Aspect Dependent Online REview Labeling for Review Generation

Parisa Kaghazgaran, Jianling Wang, Ruihong Huang, James Caverlee
Texas A&M University
College Station, Texas
{kaghazgaran,jlwang,huangrh,caverlee}@tamu.edu

ABSTRACT

Online reviews play a critical role in persuading or dissuading users when making purchase decisions. And yet very few users take the time to write helpful reviews. Encouragingly, recent advances in deep neural networks offer good potential to produce review-like natural language content. However, there is a lack of large, high-quality labeled data at both the aspect and sentiment level for training. Hence, toward enabling a writing assistant framework to help users post online reviews, this paper proposes a scalable labeling method for bootstrapping aspect and sentiment labels. Concretely, the proposed approach – Aspect Dependent Online REviews (ADORE) – leverages the underlying distribution of reviews and a small seed set of labeled data through carefully designed review segmentation and label assignment. We then show how these labels can inform a generative model to produce aspect and sentiment-aware reviews. We study the effectiveness of ADORE under various scenarios such as how end-users perceive the quality of the labels and aspect-aware generated reviews. Our experiments indicate that the proposed effective labeling process along with a regularized joint generative model lead to high quality reviews with ~90% accuracy.

CCS CONCEPTS

• **Computing methodologies** → Natural language processing.

ACM Reference Format:
Parisa Kaghazgaran, Jianling Wang, Ruihong Huang, James Caverlee. 2020. ADORE: Aspect Dependent Online REview Labeling for Review Generation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401074

1 INTRODUCTION

Online reviews play a critical role in persuading or dissuading users when making purchase decisions [8]. However, many users due to lack of inspiration, time, or language literacy may refuse to post a review. While most review platforms do not share the ratios at which their users leave reviews, anecdotal evidence suggests that significantly less than 1% of all transactions lead to a review [12].

As a result, there is a growing attention in creating new methods to help users share their opinions. In one direction, some online platforms like Airbnb require hosts and guests to write mutual reviews [16]. However, such a requirement may be an impediment to customer engagement in other platforms that feature products like movies and books. In another promising direction, new tools based on natural language generation have shown good success in some domains. For example, carefully configured templates can transform well-structured data into legible text, especially for domains with consistent format and structure like weather forecast reports [1], Olympics stories [33], and corporate earnings reports [28]. However, such methods face challenges for online reviews that typically cover a broad range of categories (e.g., apps, products, restaurants) with multiple aspects within each category (e.g., food, service, staff and so on in the restaurant domain) and diverse opinions that do not fit in a single template. Hence, recent advances in deep neural networks offer good potential to produce review-like natural language content [37].

While neural language models are trained to learn the structure and the grammar of the target language in order to produce meaningful text, intrinsically they do not generate attribute-conditioned reviews to express opinions on a specific aspect of a product or service. However, sequence-to-sequence architectures have been proposed to generate natural text conditioned on the characteristics defined by the first sequence in an end-to-end manner. To name a few, works in question answering [5], conversational modeling [32] and translation [3] adopt the paradigm of sequence-to-sequence architecture.

Despite this promising progress in sequence-to-sequence problem domains, such advances have not been explored in *aspect-aware review generation* primarily due to the data bottleneck. Learning a deep neural review generator requires large amounts of labeled data. While there are many existing collections of online reviews, very few have labels at the granularity of *aspects* (like price, food quality, or decor) and with *sentiment* associated with these aspects. Furthermore, it is unclear if incorporating such aspect and sentiment into a review generator would result in meaningful reviews.

To overcome these challenges, we explore how to make use of weak supervision to expand a small set of review segments labeled with aspects and sentiments (known as the seed set) to a large amount of unlabeled review segments demanded by data-hungry neural networks. In particular, we propose and evaluate a new framework to label and generate **a**spect **d**ependent **o**nline **re**views, named ADORE. We develop a weak labeling methodology that leverages the underlying distribution of the reviews to infer weak

SIGIR '20, July 25–30, 2020, Virtual Event, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401074

(or noisy) labels. Since users express opinion on different aspects of the target in a single review, a review cannot be labeled in whole to state a specific aspect (see Figure 1). We propose a segmentation algorithm to split a review into its topically coherent segments and aim to label the resulting segments.

By overcoming this data bottleneck, we then show how to use these labels to train a generative model as if the ground truth labels are already available. In essence, this work aims to bridge aspect-mining and generative networks to generate product reviews conditioned on a specific aspect and sentiment. The proposed joint model encodes the aspect and sentiment that guides the review generator. Moving forward, we use aspect to refer to the combination of aspect and sentiment attributes. We employ a regularization technique into the language model to enhance its performance by giving rare words a proper probability to become visible to the generator. We also utilize an attention mechanism to reinforce the impact of the aspect encoder in predicting the next word.

We thoroughly analyze the ADORE framework using Yelp restaurant reviews to understand how effective is our weak labeling methodology, what is the optimal point for review segmentation to mitigate the adverse impact of the noise-prone nature of the labeling process, how our proposed approach performs compared to the baselines, and how much training data is required to reach a stable performance. We evaluate the quality of the generated reviews through a user-based study. We employ an ablation study to evaluate the impact of regularization technique on quality of the generated reviews. Concretely, our contributions are as follows:

(1) We develop a weak labeling methodology to build an aspect-aware review dataset and evaluate the effectiveness of our approach using crowd-sourced annotation.
(2) We propose a joint model that learns to generate aspect-aware reviews in an end-to-end manner.
(3) We extensively evaluate the effectiveness of our proposed framework, including through user-based approaches.

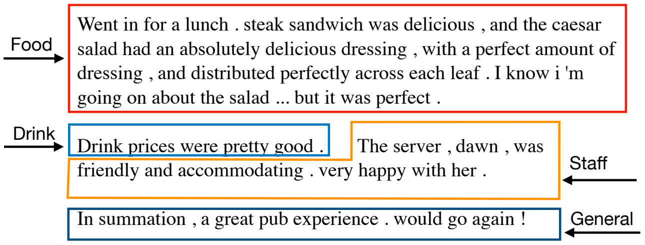
It should be noted that our aim complements efforts to improve language models. The focus of this paper is to automate labeling at the aspect-level and evaluate its success in generating high-quality aspect-aware reviews. Hence, we adapt a state-of-the-art language model [9] into our joint model for the sake of generating diverse reviews. Further, the source code of ADORE along with annotated data are available at https://github.com/Pariiiissssaaaa/ADORE_Generator

2 RELATED WORK

We discuss the related work from different dimensions and highlight our contributions.

Aspect Extraction: There is a large body of research to extract the aspects of products wherein users have expressed opinions in order to analyse the crowd sentiment towards these aspects [18, 22, 25, 30, 35]. These methods are based on relatively explicit representations of the text and attempt to model each topic as distribution of its words. This work aims to find topic boundaries so that we can segment a review into its aspect-specific parts to build the ground truth for generating aspect-aware reviews.

Text and Review Generation. Natural language generation techniques place structured data into well-designed templates [19, 27].



Went in for a lunch . steak sandwich was delicious , and the caesar salad had an absolutely delicious dressing , with a perfect amount of dressing , and distributed perfectly across each leaf . I know i 'm going on about the salad ... but it was perfect .

Drink prices were pretty good .

The server , dawn , was friendly and accommodating . very happy with her .

In summation , a great pub experience . would go again !

Figure 1: Example of Review Segmentation- A single review discusses different aspects of an item

These systems require rules and consistent format. Probabilistic approaches like N-gram models generate text by looking back only a few steps in the sequence [11, 36]. While N-gram models exhibit limitations against long text sequences, RNN-based models perform based on complex memory gating which maintain longer term dependencies [10, 21, 31]. The application of RNNs on sentiment classification of online reviews has been explored in work like [17, 26]. Yao et al. [37] propose a character-level RNN to generate fake reviews for Yelp. Due to the training methodology, these efforts cannot be easily targeted for a specific aspect and sentiment. In contrast, our work incorporates the aspect-specificity feature into the generative model.

Sequence-to-Sequence Models. Sequence models are trained to convert an input sequence into a target sequence. Applications like question answering [5], neural translation [3], chatbot [34], conversational systems [29], email auto-responses [13] and image captioning [14] have been developed under the umbrella of sequence-to-sequence architectures. However, due to lack of ground truth of aspect-aware reviews and non-sequential nature of the attributes, the direct application of seq2seq models in our problem domain is challenging.

Attribute-based Review Generation. In another direction, research has focused on generating product reviews from attributes [7, 23]. These models learn to generate customized reviews for each user based on history of their review writing. The input attributes are rating, product id and user id. In this paper, we focus on aspect as an attribute. We are aware of a car review dataset [38] at the aspect level, where each review already contains eight sentences for eight aspects and the proposed model generates reviews that cover all the aspects. However, in general domains like restaurants and e-commerce platforms, users are not forced to describe all aspects of the target. Also, they may use several sentences to describe a single aspect which we refer to as segments. In this paper, we aim to extract aspect-specific segments to build the ground truth.

3 THE PROPOSED ADORE FRAMEWORK

We propose a weak labeling methodology leveraging the underlying distribution of the reviews. Our weak labeler comprises two fundamental components: (i) *review segmentation*; and (ii) *label assignment*. Before proceeding with this approach, we pre-train a word2vec model over a large corpus of Yelp restaurant reviews (see Section 5.1) to obtain the word embeddings as w_i.

As illustrated in Figure 1, users express opinions on different aspects of the target, so a review cannot be labeled in whole to state a specific aspect. On the other hand, users typically describe an aspect in more than one individual sentence. Therefore, it is vital to detect aspect boundaries and segment reviews accordingly. In summary, the goal of review segmentation is to aggregate topically coherent *sentences* into one segment. The label assignment step aims to identify the aspect of the segments obtained from the first step using a small set of labeled data.

3.1 Review Segmentation

The review segmentation algorithm works at sentence level granularity and traverses through each review sentence by sentence in order to cluster coherent sentences into one segment. At its core, review segmentation is based on a sliding window technique with a window size of two. Each sentence is compared with the rightmost sentence in the previous segment. If their distance is less than a specific threshold τ, then the sentence is added to the segment, otherwise it forms a new segment. This process continues until the end of the review.

The segmentation algorithm is based on a metric to measure the similarity between sequential sentences. We adopt the Word Mover's Distance (WMD) [15] due to its performance to measure semantic similarity between two segments of short text. Rather than relying on keyword matching, it attempts to find an optimal transformation from one sentence S to another sentence S′ in the word embedding space:

$$\mathrm{WMD}(S_i, S_j) = \min \sum_{i}^{|S|} \sum_{j}^{|S'|} W_{ij} c(w_i, w_j) \tag{1}$$
$$\sum_{i}^{m} W_{ij} = 1/|S|, i \in \{1, ..., |S|\}, \sum_{i}^{n} W_{ij} = 1/|S'|, j \in \{1, ..., |S'|\}$$

where $|S|$ and $|S'|$ are the length of each sentence in terms of number of words and $W_{i,j}$ is the weight of word i calculated based on a normalized bag of words (nBOW) representation of a document, so it is equal to $1/|S|$ from sentence S that is transferred to word j of sentence S'. Finally, $c(w_i, w_j)$ is the traveling cost between two words and is calculated by taking the Euclidean distance between embedding representation of words.

Algorithm 1 shows the segmentation steps for one specific review R, which can then be generalized to all the reviews in the dataset. We later show how we choose the threshold empirically.

3.2 Label Assignment

Now that we split reviews into coherent segments, in this step we attempt to label the segments at the aspect level. The label assignment algorithm is based on a small set of labeled data known as the seed set. Table 1 reports the statistics of this seed set.

The main intuition is to find semantically similar seeds to the unlabeled segments and use their labels to identify the aspect of the segments. For this purpose, we compare each sample in the seed set against the unlabeled segments using the WMD distance function. If their distance is less than a threshold τ then the segments discuss similar aspects in the semantic space and so the unlabelled segment receives the same label as the seed sample. Algorithm 2 shows the label assignment steps.

Multi-label sentences. With respect to the review segmentation algorithm, the key assumption is that each sentence or several

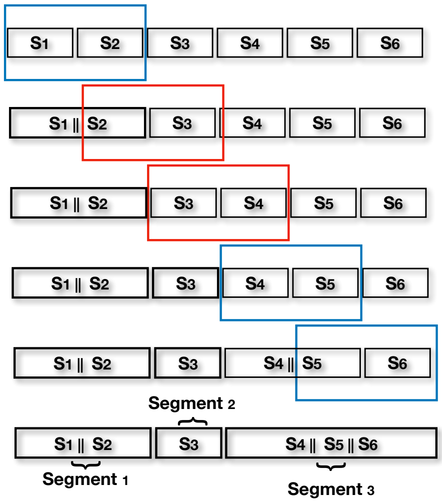


Figure 2: The review segmentation algorithm attempts to split a review into coherent segments by moving a sliding window over sequential sentences. Blue windows show two sentences are close in the semantic space, so they are clustered into one segment. Red windows shows the split point where two sequential sentence belong to different segments.

Algorithm 1 Review Segmentation

```
R : Review
S : Sentence
segments={ }
S = Split (R)
$current_{seg} = S_1$
for i = 2 to |S| do
    $d = WMD(S_i, current_{seg})$
    if $d < \tau$ then
        $current_{seg} += S_i$
    else
        $segments.add(current_{seg})$
        $current_{seg} = S_i$
$segments.add(current_{seg})$
return segments
```

Algorithm 2 Label Assignment

```
seed: labeled data
seg: unlabeled segment
for seed in $seed_{set}$ do
    for seg in segments do
        $d = WMD(S_i, current_{seg})$
        if $d < \tau$ then
            label(seg) ⟵ label(seed)
return label(segments)
```

Table 1: Distribution of labels in the seed set. (+) and (-) indicate positive and negative sentiments, respectively

Labels	Food (+)	Food (-)	Ambience (+)	Ambience (-)	Price (+)	Price (-)	Drink (+)	Drink (-)	General (+)	General (-)	Total
Count	446	163	102	16	19	24	28	6	407	261	1472
Percentage (%)	30.29	11.07	6.92	1.08	1.30	1.63	1.90	0.40	27.64	17.73	100

coherent sentences discuss only one aspect. However, expressing multiple aspects in a single sentence is a common practice when users write reviews. Table 2 shows some examples of such scenarios where there is no optimal point to divide it into individual parts corresponding to individual aspects. In this paper, we aim to deal with segments with a single aspect and leave these kinds of multi-aspect segments for future work. For this purpose, we limit our experiments to the segments that receive only one label by the label assignment algorithm.

Table 3 demonstrates examples of segments obtained from the segmentation algorithm and their labels assigned by the label assignment algorithm across different aspects and sentiments.

Table 2: Example of sentences with more than one aspect.

Multi-label Review Sentences	Labels
food is delicious only down fall is price and portion size .	**Food (+)** **Price (-)**
prices are reasonable and the food tastes great .	**Price (+)** **Food (+)**
the food was excellent and i highly recommend the business .	**Food (+)** **General (+)**
beer was all good and food generous and tasty .	**Drink (+)** **Food (+)**
highly recommend this location for quality service and price .	**General (+)** **Price (+)**

4 ASPECT-AWARE REVIEW GENERATION

Given this bootstrapping method for overcoming the data bottleneck, we now focus on a downstream task to validate the quality of these labels. Concretely, we show in this section how to extend recent generative models of text to generate aspect-aware reviews.

The generative model consists of three main components: (i) the aspect encoder; (ii) an attention-based language model; and (iii) a regularizer to enhance the performance of the generator. In the following, we explain each of these components in turn. We use aspect to refer to the combination of both aspect and sentiment.

4.1 Aspect Encoder

The first key component is an aspect encoder that constructs the embedding representation for each aspect. Given an aspect, we first map them into one-hot vectors A.

Then, we employ a fully-connected gated network to encode the input aspect as $z_i = \alpha(W.A_i + b)$, where z_i is the encoder output for aspect A_i, W and b are the weight matrix and the bias vector for the linear operation respectively and are learnable parameters. $\alpha(.)$ is the non-linear activation function chosen to be ReLU. We restrict the dimension size of the aspect vectors to be identical with the generator's hidden state since they initialize the hidden states.

4.2 Attention-based Review Generator

Basically, neural language models [6] recurrently compute hidden states that transfer the information to the next time step. At each time step t, the network takes in the current word w_t along with the current hidden state h_t, that encodes the sequence up to the time step t, and outputs a distribution over the vocabulary for the next word. The output distribution essentially describes the probability of observing each word w' in the vocabulary given the sequence $w(<= t)$ $(P(w'|w_1, \dots, w_t))$.

With this definition in mind and given the output of the aspect encoder z, our generative model learns to produce reviews based on information in z in addition to the information encoded in the hidden states h_t. For this purpose, the encoder output z initializes the first hidden states and in order to reinforce its impact throughout the network, an attention layer is introduced to capture the soft alignments between z and h_t.

Each aspect-specific review sample is a row in the input review matrix R of shape $n \times T$ where n is the number of samples in the training data and T is the number of time steps in the recurrent neural network (also interpreted as the size of back propagation through time). Since the generator aims to predict the next word at each time step, the output matrix (Y) is just like the input matrix but shifted by one word to the right defined as $Y = R[1 :]$. Without loss of generality, we now describe the architecture of the review generator at time step t. It encompasses four main layers: Embedding, stacked-GRUs, Attention, and Decoder.

The Embedding layer is a trainable matrix W_e that learns the low-dimensional representation of the input tokens. The matrix shape is defined by vocabulary size V and embedding size. We then employ a L-layer GRU recurrent network to capture the dependency among review words:

$$h_t^1 = GRU(w_t, h_{t-1}^1)$$
$$\dots$$
$$h_t^L = GRU(h_t^{L-1}, h_{t-1}^L)$$

where h_t^i is the hidden state calculated by i^{th} layer for word w_t. It should be noted that h_0^i are initialized with aspect encoder output $(h_0^1, \dots, h_0^L = z)$. In addition, the attention layer incorporates the aspect information into the hidden state calculated at the last layer for word w_t. In particular, given the hidden state h_t^L and aspect z vectors, we first apply a linear transformation to obtain a score for each vector as $s_t = (h_t^L \| z) \times W_s$, where W_s is a learnable parameter vector of shape $1 \times$ hidden size and the concatenation of two vectors

Table 3: Example of segments and their corresponding obtained from the segmentation and label assignment algorithms respectively across different aspects and sentiments. (+) and (-) indicate positive and negative sentiments, respectively.

Aspect-specific Review Segments	Label
steak sandwich was delicious and the caesar salad had an absolutely delicious dressing with a perfect amount of dressing and distributed perfectly across each leaf . i know i m going on about the salad .	**Food (+)**
today was my second visit to the place after having a good first experience but i am so disappointed with the quality of the food that i can say it has been my worst experience of food in months the sun dried tomatoes very absolutely stale to an extent that they tasted bitter the pizza base was so thick that it was uncooked and soggy the four cheese blend tasted completely different than the last time and so did the pesto sauce . no consistency with food quality .	**Food (-)**
the ambiance is nice too . it s a bit dark but they have this nice light display above on the ceiling made with mason jars . there is a comfy seating area in the bar area that s nice too .	**Ambience (+)**
however the one thing that surprised me was how dirty the restroom was in this restaurant . the floor was really dirty and toilet papers were unwell kept . the restaurant could at least have someone maintained the restroom in good shape and clean because this will reflect on how one maintains the cleanliness of the place .	**Ambience (-)**
the price is very reasonable for a family of four with plenty of leftovers to take home .	**Price (+)**
my wife i had a groupon for this place and for the price it was very poor value quality .	**Price (-)**
i had a nice glass of california cabernet . the wine list while not expansive was good . the bartender i had seemed to have a nice knowledge of what was going on with the wine that encompassed it .	**Drink (+)**
i ordered a glass of Merlot that was delivered to me in a dirty glass . the waitress was very polite and went to get me a new glass of wine but i was still unimpressed at that point .	**Drink (-)**
highly recommend for lunch . even during lunch rush it was not super packed . this would be a good place for a lunch meeting .	**General (+)**
i am not sure why anyone would like this place . the only thing it has going is location and that is simply not enough not for me .	**General (-)**

(h_t^L, z) gives a matrix of shape 2 × hidden size. The s_t determines the score for each vector. Then the attentive weight is calculated with a *softmax* function over s_t values:

$$a_t^v = \frac{\exp(s_t^v)}{\sum_v \exp(s_t^v)}$$

where v could be either a hidden state or aspect vector. a_t^v is the weight indicating the relatedness between aspect information around the next review word w_{t+1} to be predicted. Therefore, we update the hidden state in the last layer as $h_t^L = a_t^{h_t^L}.h_t^L + a_t^z.z$.

The updated h_t^L is fed into the Decoder layer. The Decoder layer is a linear transformation that decodes the hidden state to predict the next word as $o_t = W_o.h_t^L + b_o$, where W_o and b_o are the weight matrix and the bias vector for the linear operation respectively and are learnable parameters. The output vector o_t is then activated by the *softmax* function to produce the probability distribution over the words in the vocabulary $\hat{y}_t = \text{softmax}(o_t)$.

The output $\hat{y}_t$ is then compared with the ground truth y_t through cross-entropy loss as:

$$J(\vartheta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{|V|} y_{t,j}\log\hat{y}_{t,j} \tag{2}$$

Both y and ŷ are V-dimensional vectors, where V is the size of vocabulary and T denotes the length of the sequence or number of the time steps in the RNN. The network parameters are then updated over multiple iterations to minimize the loss value.

4.3 Diversity-enforcing Review Generator

Neural language models are often biased towards frequent patterns in training data while ignoring rare words. Specifically, the size of the vocabulary in the review domain is small and the generator tends to generate repeated patterns. Therefore, we employ the diversity-enforcing penalty function proposed in [9] as a regularizer to the loss function. In particular, it adds the aggregate cosine similarity between every pair of words in the vocabulary to the $J(\vartheta)$.

$$R = \frac{1}{|V|^2}\sum_{i}^{|V|}\sum_{j\neq i}^{|V|}\frac{w_i^T w_j}{\|w_i\|\|w_j\|} \tag{3}$$

Intuitively, when training the model with cross-entropy loss, at time step t the embedding of the corresponding word in the ground truth w_{t+1} is pushed to become close to the output vector o_t in order to get a larger likelihood, while the embeddings of the other words in the vocabulary are pushed towards the negative direction of o_t to receive a smaller likelihood. According to Zipf's law the word frequency in the training data is very low compared to the size of the corpus. For a concrete example, in our dataset the frequency of the popular word "is" is only about 1.22% while frequency of

Table 4: Examples of generated reviews conditioned on input label.

Input Label (Aspect & Sentiment)	Aspect-aware Generated Reviews
Food (+)	The meat is amazing and the portions are perfectly balanced.
Food (-)	The beef was dry and the chicken was tough.
General (+)	The staff is impeccable and the food is exceptional and it's not a huge thing for the money
General (-)	I would say they were extremely rude to us.
Ambience (+)	Great dining experience with a great vibe.
Ambience (-)	The room was nice, but the only thing I had was the lighting in the bathroom.
Price (+)	The prices are reasonable and the service was very good.
Price (-)	The price is a little expensive, but I would expect going for $ 20 for a steak.
Drink (+)	That was a perfect drink spot to go with a pitcher of champagne on the menu.
Drink (-)	They have a full bar with a decent selection of beers on tap, there were no descriptions of the beers on tap, but I 'm sure that's in a pint glass.

unpopular words drops drastically. Therefore, the embedding of the rare words are pushed towards negative directions of most output vectors and they get low probability in the final probability distribution $\hat{y}$. To encourage these rare words to become visible to the predictor, we need a mechanism to push their embedding toward a positive direction correlated with output vectors.

The idea of the regularization term is to increase the angle between words, i.e., minimizing the cosine similarity between any pair of word embeddings, so they are not pushed in one direction. We later in Section 5.2 show the effectiveness of the diversity-enforcing regularizer in producing diverse aspect-specific reviews.

Table 4 reports some samples of *generated* aspect-aware reviews across different aspects and sentiments.

5 EXPERIMENTS

We first describe the dataset. Then, we evaluate our proposed labeling methodology qualitatively and quantitatively. We also evaluate the quality of generated aspect-aware reviews from different angles.

5.1 Data

Seed data. This dataset originally was introduced in the *SemEval* challenge [24] in 2016 for *Aspect Based Sentiment Analysis* task. It contains 1,472 pairs of reviews and their labels after removing duplicate samples. Moving forward, we use label to refer ro the aspect and sentiment of the review. The dataset represents five different aspects as *Food, Drink, Price, Ambience and General* coupled with the two sentiments as positive and negative (10 labels in total).
Yelp Dataset. This dataset was released as part of round 13 of the Yelp challenge in January 2019: *https://www.yelp.com/dataset/challenge.* We use this dataset as the input to ADORE consisting of segmentation and labeling methods and further it is used as the ground truth required by the generative model. It has about 4M reviews for about 60k restaurants with 522M tokens. We picked the top 51k most frequent words to build our vocabulary and keep only reviews with words from the vocabulary, ending up with 2,791,379 (~3M) reviews consisting of 258M tokens.

5.2 Experimental Design

Review pre-processing: The segmentation algorithm produces ~7M segments out of the initial ~3M reviews. However, not all of the segments are subjective. For example, the sentence *"We went there for lunch."* does not express any opinion. To address this issue, we filter out subjective segments using the TextBlob library https : //planspace.org/20150607 – textblob_sentiment/ that calculates polarity of the text. It outputs about ~3.5M reviews to feed to the weak labeler.
Hyperparameters. We are interested in a general model that performs robustly across different labels, so we avoid setting up a highly tuned network and choose the hyper-parameters from singleton sets. We set the number of time steps in the recurrent model with respect to the maximum review length, i.e., 70. We set input size, hidden size, learning rate, dropout rate, batch size, number of GRU layers, and optimizer to 100, 100, 0.001, 0.2, 20, 2, and Adam respectively. We then adjust the learning rate decayed by 10 every 5 epochs. We only tune the number of epochs based on the performance of the model on the validation dataset. We empirically find that the model reaches a stable performance after about 10 epochs. We also split the review samples into training, testing and validation sets with ratio of 90%, 5% and 5% respectively.
Review Generation. The model takes in the desired aspect and the starting word w_0 which we define by a special token as < sor > in training samples. At each time step t, it takes in the word predicted at the previous time step (w_{t-1}), the hidden state h_{t-1} and aspect vector to predict the distribution for the next word w_t and also updates the hidden state to h_t. By feeding w_t back to the model, it produces another probability distribution to predict the next word. The generation process continues until the model predicts the end of the review identified by a special token as < eor >.

5.3 Experimental Results and Discussion

We evaluate the ADORE framework from two dimensions: the effectiveness of the weak labeling methodology and the performance of the generative model.

5.3.1 *Evaluation of Labeling Methodology.*

We first determine the optimal threshold for segmentation. Then we compare our segmentation algorithm with existing baselines. Finally, we conduct a user study to assess the quality of labels. It should be noted that labeling evaluation is heavily based on a user study as the end goal is to evaluate how the quality of segments and labels are perceived by end users.

Determining the distance threshold (τ) for segmentation algorithm. We conduct a user study to identify the optimal point for segmentation. This value determines whether to merge two sequential sentences into one segment or split them into different segments. We empirically observe the distance between two sequential sentences varies from 1.00 to 1.35. Note that WMD is based on Euclidean distance over normalized embedding vectors so the distance value varies from 0 – two exact sentences – to the maximum value of 2. We pick 105 sample reviews and deploy the segmentation algorithm on each review using different threshold values. We ask seven expert labelers to identify the threshold that closely replicates the segmentation that would be done by human segmenters. Table 5 shows the distribution of the thresholds determined by human judges indicating value of 1.1 gives the more accurate segmentation results.

Comparison with Baselines. We study the performance of our proposed segmentation algorithm with three baselines.
Sentence-level segmentation: This method breaks down a review into its sentences and takes each sentence as one segment with the assumption that an aspect of the target is discussed in a single sentence.
Review-level segmentation: Alternatively, we consider the whole review as a single segment with this assumption that each individual review discusses only one aspect of the target.
Text segmentation based on semantic word embeddings [2]: This algorithm is based on word embeddings for text segmentation and demonstrates state-of-the-art performance for an unsupervised method and follows the similar scenario as our work. Briefly, it assumes the whole text to be one segment and divides the text into multiple segments with sentence level granularity during a recursive process. Intuitively, a segment is coherent if the similarity between its words w_i and the segment v as a whole is maximized.

However, this approach is designed to find the topic boundaries in long text such as scholarly articles. In this study, we aim to show that our proposed unsupervised segmentation algorithm outperforms the baselines when it comes to splitting a review (which is typically much shorter than a scholarly article) into coherent aspect-specific pieces. Table 6 shows a motivating example of different segmentation baselines.

There is a standard text segmentation evaluation metric known as P_k. It measures the error, so lower scores indicate higher accuracy. However, it is based on a ground-truth of segments that are not readily available in our scenario. For this purpose, we sample 100 reviews and manually split them into their segments and aim to recognize the segmentation algorithm that produces segments most closely to the segments obtained from expert evaluators. We refer to the segments belonging to one review in the ground-truth as *reference partition* and segments obtained from the segmentation algorithm as *hypothesized partition.*

Table 5: Human labelers identify 1.0, 1.1 and 1.2 as optimal thresholds for review segmentation.

Threshold (τ)	1.0	1.1	1.2	1.25	1.3	1.35
number of votes	25	31	27	12	9	1

P_k [4] is the probability of segmentation error. It takes a window of fixed size k, and moves it across the review. At each step, it examines whether the hypothesized partition is correct based on the separation of the two ends of the window. In particular, it counts the number of disagreements between the reference and hypothesized partitions. P_k is defined as:

$$P_k = \frac{1}{N-k}\sum_{i=1}^{N-k}[\sigma_{hyp}(i, i+k) \neq \sigma_{ref}(i, i+k)]$$

Where $\sigma_{ref}(i, j)$ is a binary function whose value is one if the sentences i and j belong to the same segment and zero otherwise according to ground-truth. Similarly, $\sigma_{hyp}(i, j)$ is one if sentences i and j exist in the same segment and zero otherwise according to the segmentation algorithm. In the review domain due to short texts and sentence-level granularity for segmentation, we set the window size (k) to 1. N refers to the number of sentences in the review.

We calculate the P_k for each review segmented automatically by the algorithm against the ground-truth and take the average over all reviews. Table 7 shows the performance across different segmentation algorithms. ADORE produces P_k score with a lower value compared to its alternatives indicating its effectiveness in finding the aspect boundary inside reviews.

How does automated labeling perform compared to manual labeling? Here, we are interested in investigating if our weak labeler is comparable with manual labeling. For this purpose, we set up a crowd-based user study to verify if the labels are assigned truthfully according to human readers. We post 100 surveys on Amazon Mechanical Turk (AMT) each including a guideline and a set of reviews for which we seek a label from Turkers. The guideline has two major points: (i) it shows a sample of reviews along with their labels from the seed set to provide a context on how reviews and labels are paired with each other, (ii) it asks Turkers to label the reviews through a series of multi-choice questions. We design 100 surveys each with 10 reviews to cover all the labels. Each unique survey is assigned to three workers, i.e., 3 HITs (Human Intelligence Task) per task, giving us a total of 300 surveys and 3,000 questions.

To ensure the quality of responses, we insert a trivial question into each survey, which asks the Turker to check if a mathematical equation is False or True. It helps to manage the risk of blindly answered surveys. Furthermore, we only accept surveys from Turkers with approval rating of at least 95% and those who dwell on the survey for at least 7 minutes. We also restrict our tasks to workers located in the United States to guarantee English literacy.

Table 8 demonstrates the performance of the ADORE labeling process against human judgment across various labels. The key point is that the majority of the labels are found accurate by human evaluators with at least 80% accuracy. However, the accuracy for labels Price/negative and Drink/negative is relatively low and we

Table 6: An review sample is segmented at review-level and sentence-level and by state of the art text segmentation algorithm, our proposed ADORE algorithm and human expert labelers. ADORE segmentation is the closest to replicating human segmentation.

Segmentation Method	Segments
Review-level **(one segment)**	dinner was fantastic . service was great we started with the corn soup and the tuna tartare . we shared the filet and scallops . both delicious entrees . we did not realize the steak came with potatoes and ordered two sides mac and cheese and shishito peppers . i thought the peppers were really hot but i m a whimp i guess . we will definitely come back .
Sentence-level **(seven segments)**	dinner was fantastic .
	service was great we started with the corn soup and the tuna tartare .
	we shared the filet and scallops .
	both delicious entrees .
	we did not realize the steak came with potatoes and ordered two sides mac and cheese and Shishito peppers .
	i thought the peppers were really hot but i m a whimp i guess .
	we will definitely come back .
Text Segmentation [2] **(six segments)**	dinner was fantastic .
	service was great we started with the corn soup and the tuna tartare .
	we shared the filet and scallops . both delicious entrees .
	we didn t realize the steak came with potatoes and ordered two sides mac and cheese and shishito peppers .
	i thought the peppers were really hot but i m a whimp i guess .
	we will definitely come back .
ADORE **(Three segments)**	dinner was fantastic . service was great we started with the corn soup and the tuna tartare . we shared the filet and scallops . both delicious entrees .
	we did not realize the steak came with potatoes and ordered two sides mac and cheese and shishito peppers . i thought the peppers were really hot but i m a whimp i guess .
	we will definitely come back .
Ground-truth **(Three segments)**	dinner was fantastic . service was great we started with the corn soup and the tuna tartare . we shared the filet and scallops . both delicious entrees .
	we did not realize the steak came with potatoes and ordered two sides mac and cheese and shishito peppers . i thought the peppers were really hot but i m a whimp i guess .
	we will definitely come back .

can relate this to the fact that these labels do not have a significant representation in the seed set (Table 1). We aim to release the annotated reviews to contribute to the research community.

5.3.2 *Evaluation of Generative Model.*

We evaluate the proposed joint generative model in generating aspect-aware reviews from three dimensions. We first test how much labeled data is required to reach a stable performance. Second, we evaluate how end users perceive the quality of aspect-aware reviews. Finally, we do an ablation study on the impact of diversity-enforcing regularizer. Note that our aim complements efforts to improve language models and we focus on automated labeling at the aspect-level and propose to generate high-quality aspect-aware reviews as the downstream task for the labeled data.

How much labeled data is required to reach a stable performance? Although we propose to build a ground truth by an automated approach, it is critical to understand how much data is required to feed neural networks. In particular, when the target review domain for a specific label is not sufficiently large we aim to discover how much labeled data would address the data scarcity problem. For example, we observe that the distribution of the labels obtained from the weak labeling process not only is not uniform but extremely biased towards popular aspects like food. For instance, we receive ~300k samples labeled as food/positive while only 412 samples are labeled as drink/negative.

Table 7: ADORE outperforms the baselines by segmenting the reviews into their coherent topics more accurately.

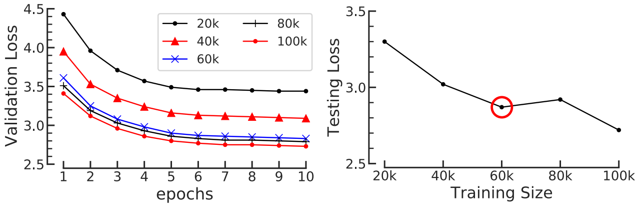
	Sentence-level	Review-level	[2]	**ADORE**
P_k	0.281	0.478	0.283	**0.265**

Table 8: Majority of the automated labels are recognized as accurate by human evaluators with > 80% acc.

Label	**Accuracy**	**Label**	**Accuracy**
Food (+)	94.33	Food (-)	85.66
General (+)	87.66	General (-)	81.00
Ambience (+)	81.33	Ambience (-)	77.66
Price (+)	86.00	Price (-)	68.00
Drink (+)	82.23	Drink (-)	57.66

Table 9: The regularized model generates more diverse reviews. TTR (Token Ratio), MTLD (Textual Lexical Diversity)

	Regularizer	Food (+)	Food (-)	Ambience (+)	Ambience (-)	Price (+)	Price (-)	Drink (+)	Drink (-)	General (+)	General (-)	All labels
Word count	✓	996	1126	1122	1422	575	995	998	1600	888	1351	11073
	×	986	1096	1090	1289	601	1136	951	1431	983	1424	10978
Unique terms	✓	**186**	**174**	**162**	**306**	45	**168**	**162**	**244**	149	**249**	**936**
	×	165	162	130	247	**51**	164	139	202	**175**	253	833
TTR	✓	**0.18**	**0.15**	**0.14**	**0.21**	0.08	**0.16**	**0.16**	**0.15**	0.16	**0.18**	**0.08**
	×	0.16	0.14	0.11	0.19	0.08	0.14	0.14	0.14	**0.17**	0.17	0.07
MTLD	✓	**65**	60	**65**	**149**	**28**	**85**	**79**	**117**	77	**150**	**81**
	×	55	60	46	122	24	84	71	96	**95**	137	71



(a) Validation loss in training

(b) Testing loss w.r.t data size

Figure 3: Stable performance with at least 60k samples.

Table 10: Majority of the generated aspect-aware reviews are perceived as reliable by human evaluators with 90% acc.

Label	Accuracy	Label	Accuracy
Food (+)	93.07	Food (-)	97.69
General (+)	86.15	General (-)	85.38
Ambience (+)	97.69	Ambience (-)	90.76
Price (+)	90.00	Price (-)	83.07
Drink (+)	90.76	Drink (-)	33.84

For this purpose, we plot the validation loss during training epochs with various training size in Figure 3a. We gradually increase the training size starting with 20k reviews. This shows that approximately 60k samples are required to improve the performance at its finest. Figure 3b plots testing loss versus the training size and echoes the same intuition.

How are aspect-aware generated reviews perceived by end users? Regardless of the model performance with respect to labeling confidence and training size, the real test is to evaluate how reviews are perceived by human readers. Similar to label assessment, we launch a crowd-based user study by posting surveys on AMT. We follow similar guidelines to those mentioned in the previous section to ensure the quality of the answers.

We design 100 surveys each with 10 generated reviews at various aspects. We assign 3 HITs per task, giving us a total of 300 surveys and 3,000 questions. We ask Turkers to label the model-generated reviews through a multi-choice questions based on the aspect.

From Table 10, we observe that generated reviews stay with the desired aspect with higher than 90% accuracy for a majority of the labels. For example, 93% and 97% of reviews on Food (positive and negative) are perceived equally by the model and the human evaluators while this number is 34% for drink/negative. We can relate this to the fact that the label *Food* has a better representation in both seed set Table 1 and our expanded dataset as it is the main topic of discussion when writing a review for a restaurant.

How does the regularized language model improve diversity? Here, we are interested to evaluate the impact of regularization on the quality of the generated reviews in terms of their diverse vocabulary. To do this, we develop the model in the presence and the absence of the optimization term described in 4.3. The testing loss is 2.80 and 2.93 respectively indicating that regularized model adapt with significantly more diverse patterns. However, we take one step further and evaluate this feature qualitatively as well.

We pick a sample of 1k reviews generated by each of the models (2k in total) across various labels. We then concatenate the samples in the same label category and report the number of words w, the number of unique terms t, and the ratio of the tokens (TTR) calculated as t/w in Table 9. The main intuition is that the more diverse reviews have a larger number of unique terms and higher values for TTR. However, in the review domain, the number of unique words are limited and TTR falls as we add more samples. Therefore, the MTLD (Textual Lexical Diversity) metric is proposed [20] that counts the number of words before TTR falls below a given threshold. By setting the threshold to 0.5, we see from Table 9 that MTLD for the regularized model is higher indicating they have more diverse vocabulary such that the more number of words are needed to meet the TTR threshold. We also report the diversity metrics for aggregation of the reviews across all the labels.

6 CONCLUSION AND FUTURE WORK

We have explored how to make use of weak labels in order to generate aspect-specific online reviews when sufficient data required by neural networks are not available. The main intuition is to: (i) build a ground truth of aspect-specific reviews automatically, (ii) propose a generative model that produces reviews conditioned on input aspects; and (iii) evaluate the quality of the labels and generated reviews. Our results are promising and, in our ongoing work, we aim to study the performance of other neural architectures to encode aspect and sentiment and expand our coverage for different review platforms. On the downside, this method could be abused for fake review generation, and so further research is needed to counter the potential of such attacks. However, since our aim is to facilitate writing authentic reviews, such a service could be limited to users who have already made a purchase on the review platform.

Acknowledgement. This work was supported in part by NSF grant SaTC-1816497. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

REFERENCES

[1] Ibrahim Adeyanju. 2015. Generating weather forecast texts with case based reasoning. *arXiv* (2015).

[2] Alexander A Alemi and Paul Ginsparg. 2015. Text segmentation based on semantic word embeddings. *arXiv preprint arXiv:1503.05543* (2015).

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR* (2015).

[4] Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning* (1999).

[5] Shiqian et al. Chen. 2019. Driven answer generation for product-related questions in e-commerce. *WSDM*.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

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

[8] Wenjing Duan and et al. 2008. Do online reviews matter? An empirical investigation of panel data. *Decision support systems* (2008).

[9] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. *ICLR* (2019).

[10] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv* (2013).

[11] Dirk Hovy. 2016. The enemy in your own camp: How well can we detect statistically-generated fake reviews–An adversarial study. In *ACL*.

[12] Yelp Inc. 2011. Yelp and the 1/9/90 Rule, https://blog.yelp.com/2011/06/yelp-and-the-1990-rule, Last Access: 08/16/2019. (2011).

[13] Anjuli Kannan and et al. 2016. Smart reply: Automated response suggestion for email. In *KDD*. ACM.

[14] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

[15] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*.

[16] Dean D Lehr. 2015. An analysis of the changing competitive landscape in the hotel industry regarding Airbnb. (2015).

[17] Zachary C Lipton and et al. 2016. Capturing meaning in product reviews with character-level generative text models. *ICLR* (2016).

[18] Kang Liu, Heng Li Xu, Yang Liu, and Jun Zhao. 2013. Opinion target extraction using partially-supervised word alignment model. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

[19] Mitul Makadia. 2018. What Are the Advantage of Natural Language Generation and Its Impact on Business Intelligence?, https://www.marutitech.com/advantages-of-natural-language-generation/, Last Access: 08/14/2019. (2018).

[20] Philip M McCarthy and Scott Jarvis. 2010. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods* 42, 2 (2010), 381–392.

[21] Stephen Merity and et al. 2017. Regularizing and optimizing LSTM language models. *arXiv* (2017).

[22] Samaneh Moghaddam and Martin Ester. 2011. ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. In *SIGIR*. ACM.

[23] Jianmo Ni and Julian McAuley. 2018. Personalized Review Generation by Expanding Phrases and Attending on Aspect-Aware Representations. In *ACL*. 706–711.

[24] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*. 19–30.

[25] Ana-Maria Popescu and Orena Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*. Springer.

[26] Alec Radford and et al. 2017. Learning to generate reviews and discovering sentiment. *arXiv* (2017).

[27] Ehud Reiter and et al. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence* (2005).

[28] Paul Roetzer. 2016. How the Associate Press and the Orlando Magic Write Thousands of Content Pieces in Seconds, https://bit.ly/2HCyAiS, Last Access: 08/14/2019. (2016).

[29] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *ACL* (2015).

[30] Lei Shu, Bing Liu, Hu Xu, and Annice Kim. 2016. Lifelong-rl: Lifelong relaxation labeling for separating entities and aspects in opinion targets. In *EMNLP*, Vol. 2016. NIH Public Access, 225.

[31] Ilya Sutskever and et al. 2011. Generating text with recurrent neural networks. In *ICML*.

[32] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *ICML* (2015).

[33] WashPostPR. 2016. The Washington Post experiments with automated storytelling to help power 2016 Rio Olympics coverage, https://wapo.st/2G67Sg6, Last Access: 08/14/2019. (2016).

[34] Anbang Xu and et al. 2017. A new chatbot for customer service on social media. In *CHI*. ACM.

[35] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. *ACL* (2018).

[36] Arun Kumar Yadav and Samir Kumar Borgohain. 2014. Sentence generation from a bag of words using N-gram model. In *ICACCCT*.

[37] Yuanshun Yao and et al. 2017. Automated crowdturfing attacks and defenses in online review systems. In *CCS*.

[38] 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*. 168–177.