OrderSum: Reading Order-Aware Unsupervised Opinion Summarization

Anonymous ACL submission

Abstract

Opinion summarization aims to create a concise summary reflecting subjective information conveyed by multiple user reviews about the same product. To avoid the high expense of curating golden summaries for training, many unsupervised methods have been recently developed. Most state-of-the-art methods utilize the extracted segments following their salience ranking as pseudo labels to train a summary generator. However, the extracted salient segments can be verbose and their reading order has been long overlooked. In this paper, we propose a reading order-aware framework, OrderSum, aiming to generate *concise* and *logical* summaries. Specifically, we first formulate the segment ordering problem in pseudo labels as path-choosing and solve it using reinforcement learning. Moreover, to generate a more concise summary, we propose to encourage the generative model to skip useless words based on the token link information derived from concise sentences, which can be collected easily from massive raw reviews by considering the ratio of sentiment/aspect words. Extensive experiments demonstrate that OrderSum benefits from the awareness of reading order and the conciseness modeling, thus being more effective than existing unsupervised methods and achieving the state-of-the-art performance.

1 Introduction

The proliferation of opinions in online reviews led to the urgent need of automatically digesting multiple reviews to facilitate informed decision making. Opinion summarization is the task of automatically generating summaries for a set of opinions about a specific target (Conrad et al., 2009). Significant progress has been observed in the supervised setting (See et al., 2017; Chu and Liu, 2019), but most accurately-annotated summaries are always at high expense; thus, unsupervised opinion summarization methods, both extractive and abstractive, have drawn more attention recently.

Extractive methods (Angelidis and Lapata, 2018; Paul et al., 2010; Tian et al., 2020) extract segments from the raw reviews and select the most salient ones as the summary. These salient segments will likely include important information for summarization, however, they will also likely embed minor or even useless information. For example, as shown in the first row of Table 1, "*... and they seem to be enjoying it*" and "*... what I'm used to which is ...*" are not as important as other information in the extracted summary. Therefore, extractive methods typically suffer from the verbose issue.

Abstractive methods (Chu and Liu, 2019; Brazinskas et al., 2020b) mainly utilize the extracted segments following their salience ranking as pseudo labels to train a summary generator. This avoids directly using the extracted segments as summary and possibly alleviates the verbose issue. However, the reading order of these segments has been long overlooked. Therefore, it remains hard for them to generate a logical opinion summary considering the semantic relationships between all the review sentences of a single product. For example, as shown in Table 1, texts such as "*I would recommend it to all*" should be better if it is not be in the middle of the summary. Otherwise, the fluency and logic of the summarization would be harmed.

To address the above problems, we propose a novel method OrderSum, which focuses on generating concise summaries in an order-aware manner. We go beyond the popular design of using the extracted salient segments as pseudo labels to train a summary generator, and further refine the pseudo labels. We formulate the summary ordering problem as a path-choosing problem — the starting point is empty, the ending point is an ordering of segments, and the action is to choose which segments should be placed next. It is popular to apply reinforcement learning for this kind of problem. Specifically, we use policy gradient algorithm and

Extractive: It fits nicely on their deck **and they seem to be enjoying it**. **I would recommend it to all**. They have numerous **pool parties and according to my daughter**, this works great. Very different from **what I'm used to which is** a regular freezer ice maker. perfect to have **for entertaining**. You just have to remember to empty it when not in use and keep it clean.
Order-unaware Generative: It fits nicely on their deck **and enjoy it**. **I would recommend it to all**. They have numerous pool parties and this works great. Very different from a regular freezer ice maker. It is perfect to have for entertaining. Remember to empty it when not in use and keep it clean.
Our OrderSum:I would recommend it to all. It fits nicely on their deck. They have numerous pool parties and this works great. It is perfect to have for entertaining. Very different from a regular freezer ice maker. It has never stopped or had any kind of problem
Human Annotation:Awesome ice maker that is easy to use, makes ice quickly, and is much more reliable than most ice makers built into refrigerators. Works well for parties and entertaining, RV traveling, as well as to save money not needing to buy ice from the store. It will eventually pay for itself!

Table 1: Real examples of extractive and order-unaware generative summaries. For a fair comparison, they are produced by the ablation versions of OrderSum, OrderSum-no-Abstractive and OrderSum-Order-unaware (see Section 3.2). Potential **verbose issue** is marked in **blue** and potential **ordering issue** is marked in **red**.

find that ROUGE-L score is ideal to be used for the reward by its definition and experimental results. We further exploit the aspect and sentiment words extracted from the raw review corpus to construct a pool of concise review sentences by considering the ratio of aspect and sentiment words. When training the generator, we incorporate the token link information derived from the concise review pool to encourage the generator to ignore useless words and output more concise summaries.

Our contributions are summarized as follows.

- We are the first to rectify the order of extracted salient segments in pseudo labels. Specifically, we formulate this problem as path-choosing and solve it using reinforcement learning.
- We propose to retrieve concise reviews from raw review corpus and then derive token link information to encourage the summary generator to ignore useless words.
- We have conducted extensive experiments on benchmark datasets, which show the superiority of OrderSum over the state-of-the-art methods.

2 Our OrderSum Method

As shown in Figure 1, our method has three key components: (1) *pseudo-label initialization by salient segments extraction*. As the first step, we obtain the aspect words and sentiment words from the review corpus, and then rank the segments of reviews according to their aspect and sentiment scores; (2) *pseudo-label refinement by segment order rectification*. To rectify the salient segments in pseudo labels, we train an ordering module with policy gradient algorithm; (3) *conciseness-aware summary generator training*. We propose to create a concise review pool from the review corpus and derive token link information from the pool. We leverage such derived link information to guide the generative model to output concise reviews.

2.1 Label Initialization: Salience Ranking

This part is not the focus of our work, but for the self-contained purpose, we briefly introduce how to extract salient segments as pseudo labels. We leave the details of deriving sentiment score and aspect score for each word in the vocabulary in Appendix. We denote $\mathcal{S}$ and $\mathcal{A}$ as the sentiment word set and aspect word set, respectively. For a word w, its sentiment and aspect scores are denoted as $\mathcal{S}(w)$ and $\mathcal{A}(w)$, respectively. All of the scores are between 0 and 1.

We mainly follow the pipeline proposed in the previous work (Angelidis and Lapata, 2018; Paul et al., 2010; Tian et al., 2020) and measure the importance of each segment in terms of sentiment polarity and aspect tendency. Specifically, given a text segment $\mathbf{x}$, we formulate its sentiment score and aspect score as follows.

$$\text{Sentiment}(\mathbf{x}) = 1+\sum_{w\in\mathbf{x}} \mathcal{S}(w); \text{Aspect}(\mathbf{x}) = 1+\sum_{w\in\mathbf{x}} \mathcal{A}(w)$$

where w refers to words in the segment and the *add-1* is designed to ensure the scores non-zero.

We then integrate these two scores using a regularized geometric mean as follows:

$$\text{RankScore}(\mathbf{x}) = \sqrt{\text{Sentiment}(\mathbf{x})^{\lambda} \cdot \text{Aspect}(\mathbf{x})}$$

where λ is a constant hyperparameter that unifies the scale of the sentiment score and aspect score, making the standard deviation of two distributions the same. Note that λ is automatically decided for

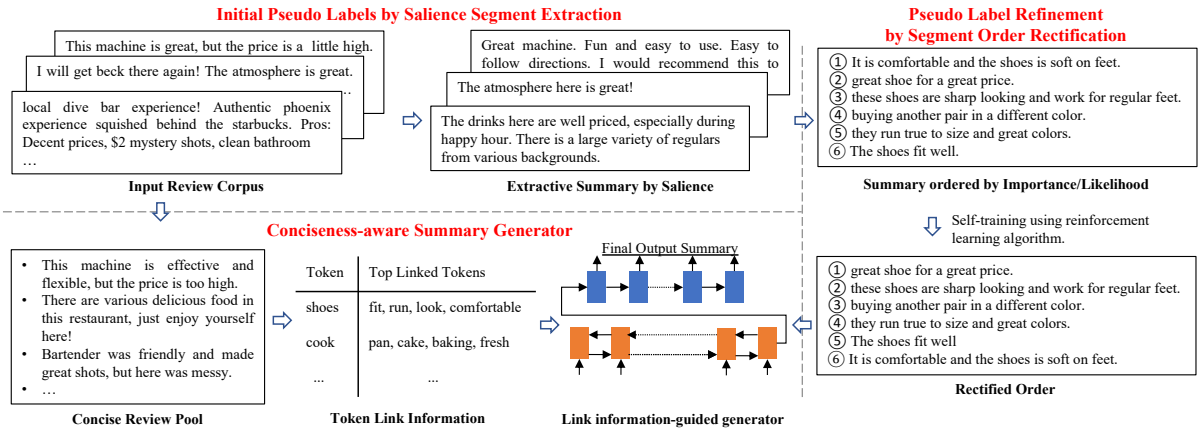

Figure 1: An overview of OrderSum. It contains three key components, (1) pseudo-label initialization by salient segments extraction, (2) pseudo-label refinement by segment order rectification, and (3) conciseness-aware summary generator training. The final output of OrderSum is generated by the summary generator.

every dataset after observing the two distributions $\{\text{Sentiment}(\mathbf{x})\}$ and $\{\text{Aspect}(\mathbf{x})\}$; no manual tuning is required. The higher the $\text{RankScore}(\mathbf{x})$ is, the more likely a segment x is considered as important. For each review set of a single product, the top-10 segments with the highest rank scores will be selected, and they will be ordered by the rank score to form the *initial pseudo labels*.

2.2 Label Refinement: Order Rectification

In this component, we will rectify the order of segments in the pseudo labels.

Ordering is Important. The order of segments in the extracted summary has been long overlooked by existing opinion summarization methods. Rouge scores (ShafieiBavani et al., 2017) are widely used criteria for opinion summarization. They measure the similarity between the gold summaries and the generated summaries. We experiment by randomly shuffling the extracted segments and observe how the performance changes. By definition, the R-1 and R-2 scores[1] focus more on the contents within a single segment, and thus should remain the same; on the other hand, the R-L score captures the ordering of consecutive segments. In our experiments, the R-L of best ordering and that of worst ordering can fluctuate by about 5-8%. Given such a significant R-L score difference, the ordering of these extracted segments should have significant impacts to the final opinion summary performance.

Self-teaching Procedure. We have designed a self-teaching procedure to rectify the order. It starts with our salient segment ranking order, which is typically adopted in existing unsupervised methods. Intuitively, this ordering should outperform most of the randomly shuffled summaries. We have confirmed this intuition by experiments on **Yelp** dataset — the R-L score of randomly shuffled summarizations will be about 5% lower than the default ordering ones. It makes sense because human-written summaries will also consider to place the more important segments in the beginning. Therefore, our self-teaching procedure starts from this ordering.

The segment ordering is never a simple ranking problem because of its context-dependent nature. For example, it is not ideal to place a segment "*this picture has amazing coloring*" between two other segments describing the prices of this picture.

We utilize policy gradient (PG) (Lin and Zhou, 2020) here to resort the order. The whole training procedure is shown in Algorithm 1. In this method, the neural network receives the input of current generated summary and the leftover segments, encode them with several fully-connected layer, outputs the softmax probabilities of choosing each next segment. The neural network will be trained by

$$\theta_{t+1} = \theta_t + \alpha R_t \bigtriangledown_\theta \ln \pi(a_t|s_t;\theta), \qquad (1)$$

where θ is the model parameter and R_t denotes the increase or decrease of R-L between current ordered summary and "gold" summary after choosing the t^{th} segments. In the beginning, the default ordering of segments (i.e., salience ranking) will be used as "gold" summary to train the policy network with PG. In each epoch of self-teaching stage, we update the "gold" summary by the new ordered

[1]The definitions of Rouge-N (R-1 and R-2) and Rouge-L (R-L) scores are given in Eq. **??**.

Algorithm 1 Ordering algorithm

Input: Generated segment set F for all test test reviews
Parameter: Reinforcement Neural Network A(θ)
Output: Generated Summary S

1: Initialize the training dataset T as the default ordering of segments in F
2: Train the Action Network with PG (see Eq. 1)
3: **while** epoch $<$ MAX_EPOCH or T keep changing **do**
4: Update the training dataset T according to Eq. 2
5: Add turbulence into the training dataset T
6: Train the Neural Network A with PG: $\theta_{t+1} = \theta_t + \alpha R_t \bigtriangledown_\theta ln\pi(a_t|s_t;\theta)$
7: **end while**
8: **return** T

summary by policy network $A(\theta)$ of product reviews. Specifically, we reconstruct the training dataset T by the following equation.

$$T_t = \{< F, \underset{\text{order}(F)}{\arg\max}\, \text{R-L}(\text{order}(F)) >\} \quad (2)$$

where T_t is the new training dataset T in the t^{th} epoch, F refers to the salient segment set for each product as the initial pseudo label, and the $\arg\max$ part computes a new "gold" ordering label for F towards the maximum R-L score.

Turbulence. To prevent the reinforcement learning process from simply overfitting the original salience ranking or the best ordering from the previous epoch, we have added some randomness in the training process. Specifically, in the 5^{th} line of Algorithm 1, we randomly choose part of the training dataset T_t and exchange segments to distort the "gold" order for each product. In this way, the training process will not simply overfit the original salience ranking.

The self-teaching training procedure will be stopped once the labels almost remain the same after each epoch, or the number of epochs reaches the maximum limit.

2.3 Conciseness-aware Summary Generator

In this component, we aim to alleviate the verbose issue caused by extractive pseudo labels. To guide the generator towards more concise summaries, we propose to identify concise review segments and derive token link information. Such token link information is further utilized to refine the pseudo labels and also guide our generator to be more concise.

Concise Review Pool. Given the huge volume of raw reviews, the quality of these reviews usually vary significantly. Therefore, before we construct the concise review pool, we filter some obvious low-quality reviews. When the average length of single word in a review is too long or too short, this review will not be considered because it's likely there exist too many informal words or messy usage of whitespace/punctuation.

We further filter the reviewers who usually write low-quality reviews. Specifically, we rank all the reviewers in the corpus according to the ratio of their reviews filtered by the aforementioned rule. In our concise review pool, we only consider the reviews from those top 30% ranked reviewers. After these two filter steps, the quality of review pool is improved greatly.

We then extract concise reviewers from the filtered review pool. We define the conciseness of a product review as the ratio of sentiment words and aspect words in the whole review texts. The top 10% reviews with the highest ratios form the concise review pool.

Token Link Information Derivation. We design a function to measure the relatedness between tokens from the concise review pool. The basic intuition behind is that if two words co-occurs more frequently in the concise review pool, it is more likely that they should be generated in the near region and contains little useless information. More specifically, the summarization should focus less on those words or phrases if the meaning of the whole sentence will basically remains the same even if they are removed. After stopwords are filtered, we use $P(a \rightarrow b)$ to represented the probability of word a occurs just within a size-s window after b in the concise review pool. Specifically,

$$P(a \rightarrow b) = \frac{\#(a \rightarrow b)}{\sum_w \#(a \rightarrow w)}, \quad (3)$$

where $\#(a \rightarrow b)$ denotes the number of times b occurs within a size-s window after a, and w refers to all the words which occurs more than σ times after word a within a size-s window. Here, σ plays a role of minimum support number to filter out too rare cases. s is typically a small value to set a proper context for consideration. In our experiments, we set $\sigma = 5$ and $s = 3$ for both datasets. It is by no means that these values are

optimal — we only want to showcase that token link information derived in this way can improve the conciseness of the generated summary.

Guided Summary Generation. The ordered segments will be input as a whole into the generator, and the output will be a paragraph of review summarization. The refined pseudo labels will be processed here to cut off some extra information. Specifically, we go through every pseudo-label summary from left to right, and at each position, check if the top-ranked tokens with highest link information score with current token exist in the near area. If so, we directly jump to that position and ignore the texts in between.

For the generator, we propose a variant of sequence-to-sequence LM in (Dong et al., 2019). The encoder and decoder and implemented by a single Bert (Devlin et al., 2018). The input is represented as $[CLS]x_1x_2x_3 \ldots x_n[SEP]y_1y_2 \ldots y_m[SEP]$, where x_i denotes the i^{th} token in the input texts, y_i denotes the i^{th} token in the generated summarization.

The Bert maps a sequence of tokens in $\mathbf{x}$ to a sequence of continuous hidden representations $(\mathbf{h}^1, \ldots, \mathbf{h}^{|\mathbf{x}|})$ with self-attention mechanism where $|\mathbf{x}|$ is length of the summarization, and the Bert model then generates the target keyphrase $(y^1, y^2, \ldots, y^{|y|})$ token-by-token in an auto-regressive manner ($|y|$ denotes the number of tokens in the keyphrase). where $\mathbf{h}^t_{enc}$, and $\mathbf{h}^{t'}_{dec}$ are hidden states at time t/t' for encoder and decoder respectively; f_{enc} and f_{dec} are auto-regressive functions implemented by LSTM cells; $o^{t'-1}$ is the predicted output of decoder at time $t'-1$; and $\mathbf{c}$ is the context vector derived from all the hidden states of encoder through a non-linear function q.

In the first timestep, all the y_i are $[MASK]$ means they are masked. At the timestep t, $y_i (i > t)$ are still $[MASK]$, the prediction of y^t is determined based on a distribution over a fixed vocabulary, conditioned on the input texts and previously generated tokens as follows:

$$p_g(y^t|y^{1,\ldots,t-1}, \mathbf{x}) = f_{out}(attn(\mathbf{h}^1, ..., \mathbf{h}^{t-1}), P(y^{t-1} \rightarrow y^t))$$

where f_{out} is a softmax classifier with an attention mechanism, $attn$ is a self-attention layer. Compared to classical generator, the extra term $P(y^{t'-1} \rightarrow y^{t'})$ utilizes concise token information to guide the generator output more concise summarization, because it encourages the generator to skip useless words (e.g., stopwords).

Table 2: Statistics of datasets. Following other unsupervised opinion summarization methods (Brazinskas et al., 2020a,b), training set only contains raw reviews, only validation set and test set contains include the summary label.

Yelp	Train	Validation	Test
#review	1,016,137	240	320
#token/review	55.8	0	0
Amazon	Train	Vlidation	Test
#review	3,889,782	96	160
#token/review	65.7	49.9	49.1

3 Experiments

In this section, we conduct extensive experiments to compare our OrderSum method with many other methods on two benchmark datasets.

3.1 Datasets

We perform experiments on the benchmark Amazon dataset (He and McAuley, 2016) and Yelp dataset [2]. They both include a large training corpus of reviews for businesses without gold standard summaries. Following previous work (Brazinskas et al., 2020a), we selected 4 categories from the Amazon dataset: *Electronics; Clothing, Shoes and Jewelry; Home and Kitchen; Health and Personal Care*. In the test set of both datasets, 3 expert-written label summaries are conditioned on 8 reviews for each product.

3.2 Compared Methods

To show the superiority of our model, we compare our method with the following state-of-the-art unsupervised methods:

- **MEANSUM** (Chu and Liu, 2019) consists of an auto-encoder where the mean of the representations of the input reviews decodes to a reasonable summary-review.
- **COPYCAT** (Brazinskas et al., 2020b) models review groups with continuous latent representations, and applied novelty reduction mechanism and copy mechanism.
- **PLANSUM** (Amplayo et al., 2021) explicitly incorporates content planning that takes the form of aspect and sentiment distributions derived from data, and the synthetic datasets are created by sampling pseudo-reviews from a Dirichlet distribution parameterized by the content planner.

We have also explored four variants of our **OrderSum** as follows. (1) **OrderSum-No-**

[2] https://www.yelp.com/dataset/challenge

Table 3: Automatic and Human Evaluations on Amazon and Yelp datasets. Human Evaluation range from $1worst$ to $4best$.

	Yelp						Amazon					
Model	**R-1**	**R-2**	**R-L**	**Info**	**Read**	**Corr**	**R-1**	**R-2**	**R-L**	**Info**	**Read**	**Corr**
MEANSUM	0.289	0.037	0.159	1.88(± 0.91)	1.50(± 0.86)	1.65(± 0.94)	0.292	0.047	0.187	1.79(± 0.81)	1.96(± 0.97)	1.93(± 0.93)
COPYCAT	0.295	0.053	0.181	2.38(± 1.07)	2.76(± 0.94)	2.64(± 1.13)	0.320	0.058	0.201	2.46(± 0.95)	2.72(± 0.94)	2.66(± 1.08)
PLANSUM	0.348	0.070	0.197	2.48(± 0.87)	2.68(± 0.84)	2.59(± 0.96)	0.329	0.061	0.191	2.56(± 1.02)	2.34(± 0.86)	2.40(± 1.04)
OrderSum	**0.357**	**0.081**	**0.215**	**3.26**(± 0.89)	**3.05**(± 0.84)	**3.12**(± 0.96)	**0.339**	**0.071**	**0.211**	**3.19**(± 0.86)	**2.98**(± 0.90)	**3.01**(± 0.99)

Table 4: Ablation Study on Amazon and Yelp datasets.

	Yelp			Amazon		
Model	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
OrderSum	**0.357**	**0.081**	**0.215**	**0.339**	**0.071**	**0.211**
OrderSum-No-Absractive	0.342	0.065	0.194	0.328	0.062	0.192
OrderSum-Order-unaware	0.357	0.073	0.180	0.331	0.064	0.178
OrderSum-No-LinkInfo	0.346	0.070	0.202	0.331	0.066	0.199
OrderSum-Random	0.312	0.064	0.185	0.308	0.059	0.176

Abstractive directly uses the extracted salient segments following the rectified order as the output. This variant presents the quality of the pseudo labels after rectification. It can be also viewed as a strong extractive baseline. (2) **OrderSum-Order-unaware** skips the pseudo label rectification step and keeps the other parts the same. (3) **OrderSum-No-LinkInfo** ignores the link information and sticks to the traditional generator. (4) **OrderSum-Random** randomly choose one of the reviews as the pseudo-label.

Because most, if not all, existing extractive methods perform worse than the abstractive ones, we do not include extractive methods for comparison.

3.3 Evaluation Metrics

Automatic Evaluation. ROUGE score (Lin, 2004) is a standard summarization metric to measure the correlation between a generated summary and the reference summary. In our experiment, we use the Rouge-1 (**R-1**), Rouge-2 (**R-2**), and Rouge-L (**R-L**) for each product/business and then average them across different products/businesses as the automatic evaluation. Rouge score can avoid the hallucinating facts and entities problem in some way (Falke et al., 2019).

Human Evaluation. Following previous work (Zhang et al., 2021), we use the informativeness (**Info**), readability (**Read**) and correlations (**Corr**) between generated summaries and gold summaries as our human evaluation criteria. Specifically, human evaluation for the generated summaries is conducted to quantify the qualitative results of each model. We have hired 25 annotators to ensure a high-quality evaluation. For each dataset, we randomly select 30 products/businesses and present the summaries generated by all 4 compared baselines in a random order to each human interviewee. For a specific criterion, different summaries will be ranked by the human interviewee, receiving a score from 4 (the best) to 1 (the worst). We report the average scores for each method on each dataset.

3.4 Hyper-parameter Settings

The dimension of embeddings layer is 300, and the embeddings dropout is 0.1. We used a stack of 3 layers LSTM, the hidden dimension is 256, the batch size is 32, the maximal epoch is 30 and the dropout is 0.3. We use the Adam optimizer (Kingma and Ba, 2015) and the decayed learning rate is initialized as 0.001. In our experiments, the hyperparameter tuning was based on the Yelp and Amazon validation sets as the previous work (Brazinskas et al., 2020a). Our model was trained on a single NVIDIA A6000 GPU and is implemented using PyTorch.

3.5 Results and Discussions

Our main results are shown in Table 3. As one can see, OrderSum has achieved the best performance compared to three recent state-of-the-art unsupervised methods in terms of all the automatic and human evaluation scores. PLANSUM has the second best performance. Among R-1, R-2 and R-L scores, R-L is arguably the most difficult to improve. OrderSum has about 1.5% R-L improvement over PLANSUM, which shall be considered as significant. Among all three human evaluation criteria, the gap of informativeness score between OrderSum and other models is the most obvious.

To investigate the role of each component, three ablations of OrderSum are also compared. The performance are shown in 4. OrderSum-No-Abstractive shows relatively obvious drops in R-1

Initial Pseudo Label

It is very comfortable and the shoes is soft on feet. great shoe for a great price. these shoes are awesome. They are really sharp looking and probably would work for someone with a regular feet. I will be buying another pair in a different color. they run true to size and the colors are great. the shoes are the perfect fit for me.

Generated without link information

It is very comfortable. the shoes is soft. great shoe for a price. these shoes are really sharp looking and probably would work for regular feet. I buying another pair in a different color. they run true to size and the colors are great. the shoes are the perfect fit for me.

Generated with link information

It is comfortable and the shoes is soft on feet. great shoe for a great price. these shoes are sharp looking and work for regular feet. buying another pair in a different color. they run true to size and great colors. The shoes fit well.

Figure 2: Example summaries for a product in the Amazon dataset, investigating the influence of link information.

and R-2, because extracted salient segments typically carry some redundant information, thus seriously influencing the number of unmatched unigrams and bigrams. Also, the informativeness of OrderSum-No-Abstractive is close to that of OrderSum, its readability and correlation scores are far worse than OrderSum. It means the abstractive stage did not influence the main content of the generated summary, but improved the readability much by cutting off those redundant texts.

There is not much gap between the R-1 and R-2 scores of OrderSum-Order-unaware and that of OrderSum, because the ordering step mainly influence the R-L score according to our previous analysis. Without the segment order rectification step for pseudo labels, the R-L score is only better than MEANSUM. This demonstrates the importance and necessity of our order rectification step.

OrderSum-No-LinkInfo does not utilize token link information in the abstractive stage, so the R-2 score is between that of OrderSum and OrderSum-No-Abstractive. OrderSum-No-LinkInfo has the highest R-L score except OrderSum, which means the link information will bring minor influences to the logic flow of generated summary.

3.6 Case Studies: Conciseness in Summaries

Token link information is derived from the raw review corpus and designed to improve the conciseness of generated summaries. In this section, we present some case studies.

Table 5 presents link information for three example tokens from Amazon dataset. One can easily find the top-ranked link tokens are closely related to the tokens of interest.

Figure 2 presents a case about how the link information would affect the generated summary. Guided by the token link information, "*the shoes are perfect fit for me*" has been changed to "*the shoes fit well*", and at the same time, some auxiliary words such as "*very*" and "*probably would*" have been completely skipped. This is really making the generated summary more concise.

Table 5: Example Token Link Information on the Amazon Dataset.

Token a	Token b with top 5 $P(a \rightarrow b)$
shoes	fit, run, work, look, comfortable
cook	pans, cake, baking, fresh, breakfast
battery	life, lasts, operated, recharging, charger

3.7 Case Studies: Logic Flow in Summaries

From the entire review corpus, we expect the model to learn the general and natural logic flow of an opinion summarization.

Figure 3 shows the generated summaries by OrderSum and its variants for a certain product in the Amazon dataset. In this case, we argue that it is better to place the overall comment such as "I would recommend it" in the beginning of the summary; also, sentences describing the same aspect of the product (e.g., "*easy*") should be next to each other. The expression logic in the summary generated by OrderSum-Order-unaware is not very smooth. It will jump between different aspects of a product randomly, which increase the difficulty for the reader to quickly grasp the key message.

4 Related Work

Existing work on unsupervised opinion summarization can be categorized into extractive methods and abstractive methods.

Extractive methods tries to select salient segments from the input reviews. Most of them (Angelidis and Lapata, 2018; Paul et al., 2010; Tian et al., 2020) assign sentiment polarity to each segment, then induce aspect labels from raw texts or a small number of gold summaries, finally design a heuristic function or a clustering method to construct the opinion summarization. Nishikawa et al. (2010) also focuses on the ordering phrase by linear in-

OrderSum

I would recommend this! Love this tortilla maker. It is a great machine. Fun and easy to use. Easy to follow directions. Probably great and easy for tortillas. Perfect tortillas every time. Tortillas taste so fresh.

OrderSum-no-Abstractive

I would recommend this to anyone looking for an electric tortilla maker! Great machine. Love this tortilla maker. Probably great and easy for tortillas. Fun and easy to use. Easy to follow directions. Tortillas taste so fresh. I use it 3-4 times a week and perfect tortillas every time.

OrderSum-Order-unaware

It is a great machine. Fun and easy to use. Easy to follow directions. I would recommend this! Tortillas taste so fresh. Love this tortilla maker. Perfect tortillas every time. Probably great and easy for tortillas.

Figure 3: Example summaries for a product in the Amazon dataset, investigating the influence of ordering.

teger programming, but it is a supervised method. The performance of this type of methods is typically limited by the quality of aspect words and sentiment words. Also, as shown in our analyses and experiments, the selected segments are usually redundant and the salience ranking order does not necessarily match the best logical order, which can both hurt the summarization performance.

Abstactive methods are recently more popular and their performance are better than pure extractive ones. Abstractive methods (Chu and Liu, 2019; Brazinskas et al., 2020b) are trained on large collections of unannotated product reviews, attempting to model the prevalent opinions in the input reviews and generate texts that articulate them. Chu and Liu (2019) directly inputs the average embeddings of reviews in to the decoder to generate opinion summarization. There are several recent methods incorporate other resources into the unsupervised opinion summarization problem, such as the aspect and sentiment distributions in the encoding stage (Amplayo et al., 2021) and topic-tree structure (Isonuma et al., 2021). Wang and Wan (2021) utilizes contrastive learning to learn the crucial aspect and sentiment embeddings of reviews. They are considered to be more informative and less redundant than pure extractive methods, as confirmed in our experiments. OrderSum improves over existing abstractive methods by rectifying the segment order in pseudo labels and also incorporating the token link information. These techniques as shown in our ablation study can improve the logical order and the conciseness of the generated reviews.

Our work is also closely related to reinforcement learning, which is very popular recently and achieves great performance in many areas (El-Laham and Bugallo, 2021). The policy gradient method is classical and easily used in many problems (Paternain et al., 2021) once the state space, action space and reward are set properly. This method is very suitable to solve problem such as path planning (Cui and Wang, 2021; Sang et al., 2021), so once we transform the ordering stage of OrderSum into a similar problem, it is natural to apply this technique.

Copy mechanism (Meng et al., 2017) is widely used for opinion summarization problem (Brazinskas et al., 2020b). When generating the summarization, copy mechanism will focus more on the token existed in the input texts. The way we use token link information is similar to copy mechanism, however, our method has some unique advantages. First, copy mechanism will only influence the predicted probability of tokens existed in the input texts, while our method will influence the tokens out of the input texts as long as they exist in the raw review corpus and are suitable for the logical flow. Second, copy mechanism did not explicitly utilize the the whole raw review corpus for each input, but the extra term in our method are computed from the statistics of the whole raw review corpus.

5 Conclusions and Future Work

In this paper, we propose a novel method OrderSum for unsupervised opinion summarization problem. OrderSum mainly improves the summary quality by rectifying the segment order in the pseudo labels and encouraging conciseness in the generator training. From our extensive experiments, based on both automatic and human evaluations, we have demonstrated that the superiority of OrderSum over state-of-the-art methods as well as the importance and necessity of each individual component of OrderSum. In the future, we will explore to fuse the ordering stage into the generating stage, making it an end-to-end model. Besides, the token link information can also be improved by dynamically adjustment in the training stage.

6 Ethical Considerations

The datasets used in our experiments are open resources on Internet; our work can be applied to extract concise review summary from a large number of reviews on the websites. They seem to be low-risk applications for us; we also avoid 'attributing identity characteristics' in our work; our work doesn't require much computing resources, so there is not much energy consuming.

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Unsupervised opinion summarization with content planning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications ofm Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12489–12497. AAAI Press.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3675–3686. Association for Computational Linguistics.

Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2020a. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4119–4135. Association for Computational Linguistics.

Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2020b. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5151–5169. Association for Computational Linguistics.

Eric Chu and Peter J. Liu. 2019. Meansum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 Jmune 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR.

Jack G. Conrad, Jochen L. Leidner, Frank Schilder, and Ravi Kondadadi. 2009. Query-based opinion summarization for legal blog entries. In *The 12th International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 8-12, 2009, Barcelona, Spain*, pages 167–176. ACM.

Zhengyang Cui and Yong Wang. 2021. UAV path planning based on multi-layer reinforcement learning technique. *IEEE Access*, 9:59486–59497.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Yousef El-Laham and Mónica F. Bugallo. 2021. Policy gradient importance sampling for bayesian inference. *IEEE Transactions on Signal Processing*, 69:4245–4256.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2214–2220. Association for Computational Linguistics.

Ruining He and Julian J. 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. ACM.

Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2021. Unsupervised abstractive opinion summarization by generating sentences with tree-structured topic guidance. *CoRR*, abs/2106.08007.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. page 10.

Kaixiang Lin and Jiayu Zhou. 2020. Ranking policy gradient. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada,*

July 30 - August 4, Volume 1: Long Papers, pages 582–592. Association for Computational Linguistics.

Hitoshi Nishikawa, Takaaki Hasegawa, Yoshihiro Matsuo, and Genichiro Kikui. 2010. Opinion summarization with integer linear programming formulation for sentence extraction and ordering. In *Coling 2010: Posters*, pages 910–918, Beijing, China. Coling 2010 Organizing Committee.

Santiago Paternain, Juan Andrés Bazerque, Austin Small, and Alejandro Ribeiro. 2021. Stochastic policy gradient ascent in reproducing kernel hilbert spaces. *IEEE Transactions on Automatic Control*, 66(8):3429–3444.

Michael Paul, ChengXiang Zhai, and Roxana Girju. 2010. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 66–76, Cambridge, MA. Association for Computational Linguistics.

Qiming Sang, Yu Tian, Qianlong Jin, and Jiancheng Yu. 2021. A path planning strategy for marine vehicles based on deep reinforcement learning and data-driven dynamic flow fields prediction. In *6th International Conference on Automation, Control and Robotics Engineering, CACRE 2021, Dalian, China, July 15-17, 2021*, pages 466–471. IEEE.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond K. Wong, and Fang Chen. 2017. A semantically motivated approach to compute ROUGE scores. *CoRR*, abs/1710.07441.

Yufei Tian, Jianfei Yu, and Jing Jiang. 2020. Aspect and opinion aware abstractive review summarization with reinforced hard typed decoder. *CoRR*, abs/2004.05755.

Ke Wang and Xiaojun Wan. 2021. TransSum: Translating aspect and sentiment embeddings for self-supervised opinion summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 729–742, Online. Association for Computational Linguistics.

Xinyuan Zhang, Ruiyi Zhang, Manzil Zaheer, and Amr Ahmed. 2021. Unsupervised abstractive dialogue summarization for tete-a-tetes. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14489–14497. AAAI Press.