Noisy Pairing and Partial Supervision for Stylized Opinion Summarization

Hayate Iso
Megagon Labs
hayate@megagon.ai

Xiaolan Wang[*]
Meta Platforms, Inc.
xiaolan@meta.com

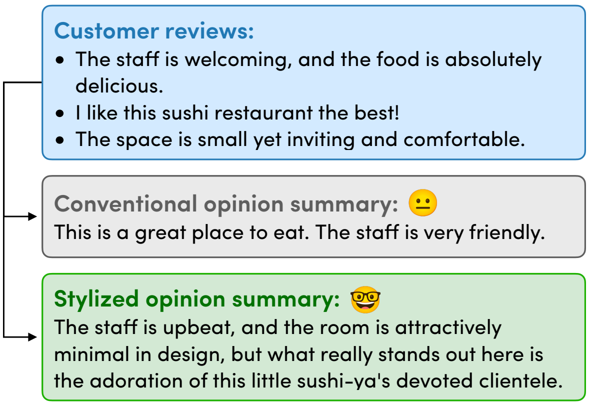
Yoshi Suhara[*]
NVIDIA
ysuhara@nvidia.com

Abstract

Opinion summarization research has primarily focused on generating summaries reflecting important opinions from customer reviews without paying much attention to the writing style. In this paper, we propose the stylized opinion summarization task, which aims to generate a summary of customer reviews in the desired (e.g., professional) writing style. To tackle the difficulty in collecting customer and professional review pairs, we develop a non-parallel training framework, Noisy Pairing and Partial Supervision (*Napa*🍷), which trains a stylized opinion summarization system from non-parallel customer and professional review sets. We create a benchmark ProSum by collecting customer and professional reviews from Yelp and Michelin. Experimental results on ProSum and FewSum demonstrate that our non-parallel training framework consistently improves both automatic and human evaluations, successfully building a stylized opinion summarization model that can generate professionally-written summaries from customer reviews.[1]

1 Introduction

Opinion summarization, which focuses on automatically generating textual summaries from multiple customer reviews, has received increasing attention due to the rise of online review platforms. Different from single-document summarization tasks (e.g., news summarization), which can easily collect a large amount of document-summary pairs, manually creating summaries from multiple reviews is expensive; it is not easy to collect large-scale training data for opinion summarization. To address this challenge, existing studies build pseudo-reviews-summary pairs in a self-supervised fashion (Chu and Liu, 2019; Amplayo and Lapata, 2020; Suhara et al., 2020; Amplayo et al., 2021; Iso et al., 2021) or use a small amount of reviews-summary pairs in a few-shot manner (Bražinskas et al., 2020a; Oved and Levy, 2021; Iso et al., 2022) to train opinion summarization models.



Customer reviews:
- The staff is welcoming, and the food is absolutely delicious.
- I like this sushi restaurant the best!
- The space is small yet inviting and comfortable.

Conventional opinion summary: 😐
This is a great place to eat. The staff is very friendly.

Stylized opinion summary: 🤓
The staff is upbeat, and the room is attractively minimal in design, but what really stands out here is the adoration of this little sushi-ya's devoted clientele.

Figure 1: Comparison of conventional and stylized opinion summarization. Given multiple reviews as input, stylized opinion summarization aims to generate a summary in the desired writing style.

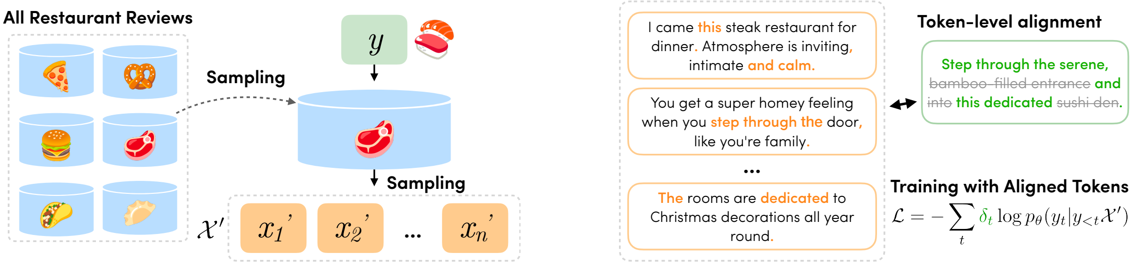
However, existing opinion summarization systems have focused on summarizing important opinions in reviews while not paying much attention to the writing style. They leverage customer reviews as pseudo summaries to train models, which generate summaries in the same writing style as the customer reviews as illustrated in Figure 2. On the other hand, professional reviews, such as Michelin Guide—a prestigious and popular restaurant guide, use a quite different writing style to describe the same type of information.

In this paper, we aim to fill this gap between customer and professional reviews by proposing a new branch of opinion summarization—*stylized opinion summarization*, where the goal is to generate a summary of opinions in the desired writing style. Specifically, besides customer reviews, as the input to the conventional opinion summarization task, we use a few example summaries in the desired writing

[*] Work done while at Megagon Labs.

[1] The code is available at https://github.com/megagonlabs/napa

Proceedings of the 17th International Natural Language Generation Conference, pages 13–23
September 23–27, 2024. ©2024 Association for Computational Linguistics


All Restaurant Reviews
Sampling
y
Sampling
$\mathcal{X}'$
x_1'
x_2'
...
x_n'
I came this steak restaurant for dinner. Atmosphere is inviting, intimate and calm.
You get a super homey feeling when you step through the door, like you're family.
...
The rooms are dedicated to Christmas decorations all year round.
Token-level alignment
Step through the serene, ~~bamboo-filled entrance~~ and ~~into~~ this dedicated ~~sushi den~~.
Training with Aligned Tokens
$\mathcal{L} = -\sum_t \delta_t \log p_\theta(y_t|y_{<t}\mathcal{X}')$

(a) **Noisy Pairing**: Given the candidate summary y, the pairs of noisy input reviews and output summary, $(\mathcal{X}', y)$, are built by retrieving the input reviews from a set of reviews from an arbitrary entity. This example retrieves the reviews from a steak restaurant given the professionally written summary of a sushi restaurant.

(b) **Partial Supervision**: After building a noisy input-output pair, we obtain the token-level alignment between the pair based on the word, stem, and synonym matching. Finally, we introduce indicator functions δ_t into the standard negative log-loss function $\mathcal{L}$ to train using only aligned tokens, highlighted in **green**.

Figure 2: Overview of our non-parallel training framework, Noisy Pairing and Partial Supervision.

style as auxiliary information to guide the model in learning the writing style. Since a few summaries in the desired writing style may not cover the same entities (e.g., restaurants) as the customer review set, the two review sets for the stylized opinion summarization task are non-parallel, which makes the task more challenging.[2]

To this end, we develop a non-parallel training framework, *Noisy Pairing and Partial Supervision* (*Napa*🍷), which builds a stylized opinion summarization model from *non-parallel* customer and professional review sets. The core idea consists of two functions: *Noisy Pairing* (§4.1) creates pseudo "noisy" reviews-summary pairs forcibly for each summary in the desired writing style by obtaining input reviews similar to the summary. Then, *Partial Supervision* (§4.2) trains a model with the collected noisy pairs by focusing on the sub-sequence of the summary that can be reproduced from the input reviews while not learning to hallucinate non-existing content. Figure 2 illustrates the two functions. In this example, for a professionally-written review of a sushi restaurant, Noisy Pairing finds reviews of a steak restaurant as noisy source reviews, which are then *partially* used by Partial Supervision to train a stylized opinion summarization model.

We also create and release a benchmark for stylized opinion summarization named PROSUM, which consists of 700 paired Yelp reviews and Michelin point-of-views. Experimental results on PROSUM confirm that *Napa*🍷 successfully generates summaries in the desired writing style in a non-parallel training setting, significantly better than models trained by self-supervision and existing non-parallel training methods.

We further performed additional experiments using existing supervised opinion summarization benchmarks, FewSum (Bražinskas et al., 2020a), in a non-parallel setting. We observed that *Napa*🍷 brings significant gains over self-supervised systems and competitive performance with state-of-the-art supervised systems, indicating the generalizability of the proposed method.

2 The PROSUM Corpus

Data Collection We build a stylized opinion summarization dataset, PROSUM, which pairs customer reviews and professional reviews about the same restaurant, as we need customer reviews as the input and a professional review as the summary for evaluation purposes.

We first collected 700 professionally-written restaurant reviews from `guide.michelin.com`, a famous restaurant review site. Unlike crowd-sourced opinion summaries, these reviews are written by professional writers. Thus, they include more appealing expressions and attractive information than crowd-sourced summaries. Then, we collected customer reviews from a popular customer review platform, `yelp.com`, by asking crowdsourced workers from Appen[3] to find the same restaurant for each of the restaurants we collected in the first step. We collected up to 5,000 customer reviews for each restaurant.

Filtering Since our main focus is to create a stylized opinion summarization benchmark and thousands of input reviews cannot be handled by most pre-trained language models, we filtered source customer reviews to reduce the number of input

[2] We will also evaluate the parallel setting later.

[3] `https://appen.com/`

	Src len.	Tgt len.	% of novel n-grams in gold summary				Extractive oracle		
			Unigram	Bigram	Trigram	4-gram	R1	R2	RL
ProSum (ours)	1162.7	139.7	38.19	84.76	97.17	99.18	42.97	10.99	22.59
Yelp (Bražinskas et al., 2020a)	453.3	58.02	31.71	83.02	95.53	98.35	47.79	15.28	25.84
Amazon (Bražinskas et al., 2020a)	446.2	56.89	31.62	82.32	95.84	98.60	46.31	14.27	25.44

Table 1: Statistics of ProSum and FewSum Yelp/Amazon benchmarks. ProSum has a longer source and target length compared to the FewSum benchmarks and offers more abstractive summaries with respect to the novel n-gram ratio. The source and target length is the number of BPE tokens per example using the BART tokenizer.

reviews to a size that can be handled by commonly used pre-trained language models.

For each reviews-summary pair, we selected source Yelp reviews so that the coverage of the target Michelin review was maximized. Specifically, we used the sum of the ROUGE-1/2 Recall scores between the selected source Yelp reviews and the target Michelin review to measure the coverage. We incrementally added source reviews until the total length exceeded 1,024 words to maximize the coverage in a greedy manner. On average, 6.7 input reviews were selected for each pair. This selection step is to ensure the target Michelin summary can be created by source Yelp reviews.

Finally, we shuffled the selected source reviews to remove the selection order bias. The final benchmark consists of 100/100/500 entities for the training/validation/test set. Note that we keep parallel data (i.e., reviews-summary pairs) in ProSum for evaluation and for training supervised models. For Napa🍷 or other non-parallel training models, we remove source reviews from the training set.

Statistics We summarize the ProSum dataset and compare it with existing opinion summarization datasets in Table 1. We calculate novel n-grams in gold summaries to evaluate how abstractive/extractive ProSum is and the performance of the extractive oracle summaries from the source reviews. We confirm that the ProSum is more abstractive than the existing benchmarks. The extractive oracle performance supports the feasibility of stylized opinion summarization in ProSum.

3 Self-supervised Opinion Summarization

This section describes the standard self-supervised framework for conventional opinion summarization and then the pseudo-reviews-summary pair construction approach (Elsahar et al., 2021), which is also used as the pre-training method in §5.

Opinion summarization is a multi-document summarization problem that aims to generate a textual summary text y that reflects the salient opinions given the set of reviews $\mathcal{X} = \{x_1, \ldots, x_N\}$. Due to the unavailability of a sufficient amount of reference summaries for training, a commonly used approach is to create a pseudo-reviews-summary training pair $(\tilde{\mathcal{X}}, \tilde{y})$ from a massive amount of reviews and trains an opinion summarization model p_θ using negative log-loss minimization,

$$\mathcal{L} = -\log p_\theta(\tilde{y}|\tilde{\mathcal{X}}) = -\sum_t \log p_\theta(\tilde{y}_t|\tilde{y}_{<t}, \tilde{\mathcal{X}}).$$

Pseudo reviews-summary pairs construction Let $\mathcal{R}_e$ denotes the set of reviews for specific entity e such as a restaurant. For each set of reviews $\mathcal{R}_e$, we treat a review in this set as a pseudo summary $\tilde{y} \in \mathcal{R}_e$ and then retrieve the relevant reviews to build a source set of reviews $\tilde{\mathcal{X}}$. Concretely, given a pseudo summary $\tilde{y}$, retrieve the source set of N reviews $\tilde{\mathcal{X}}$ by maximizing the sum of the similarity as follows:

$$\tilde{\mathcal{X}} = \underset{\mathcal{X} \subset \mathcal{R}_e \setminus \{\tilde{y}\}, |\mathcal{X}|=N}{\arg\max} \sum_{x \in \mathcal{X}} \mathrm{sim}(x, \tilde{y}),$$

where similarity is measured by the cosine similarity of the TF-IDF vector. This operation is applied to all reviews as pseudo summaries. Then the top-K pseudo-reviews-summary pairs with the highest similarity scores $\sum_{x \in \tilde{X}} \mathrm{sim}(x, \tilde{y})$ are retained as the final pseudo-training set $\{(\tilde{\mathcal{X}}_i, \tilde{y}_i)\}_{i=1}^{K}$.

4 Napa🍷

Although pseudo-reviews-summary pairs creation has been one of the solid approaches for conventional opinion summarization, we cannot directly use it for stylized opinion summarization, as there are two sets of *non-parallel* reviews in different writing styles.

This section describes a non-parallel training framework for stylized opinion summarization, *Noisy Pairing and Partial Supervision* (Napa🍷), which trains a summarization model from non-parallel customer and professional review sets.

4.1 Noisy Pairing

Noisy Pairing expands the existing pseudo-reviews-summary construction approach to create "noisy" reviews-summary pairs for each summary in the desired writing style by obtaining input reviews similar to the summary.

To leverage the desired style of summary y for the entity e, which is not paired with the set of reviews for the same entity $\mathcal{R}_e$, we first build the *noisy* reviews-summary pairs. Specifically, given the summary y for entity e, we follow the pseudo data construction approach (§3) to construct the source set of reviews, but we retrieve the reviews from the *different* entity $e'(\neq e)$ with the summary:

$$\tilde{\mathcal{X}}' = \operatorname*{arg\,max}_{\mathcal{X}\subset\mathcal{R}_{e'}, |\mathcal{X}|=N} \sum_{x\in\mathcal{X}} \mathrm{sim}(x, y).$$

For instance, given a summary of a sushi restaurant, we can use reviews of a steak restaurant to construct a noisy reviews-summary pair as illustrated in Figure 2. Then, using the similar approach used in the pseudo data construction, we obtain the final noisy training set $\{(\tilde{\mathcal{X}}', y)\}$. In particular, the top 10 noisy reviews-summary pairs of the highest similarity score are retained for each summary.

Note that this method could unintentionally select the review of the correct entity as input (i.e., $e' = e$), so in our experiments, we explicitly discarded the review of the entity used in summary to maintain the non-parallel setting.

4.2 Partial Supervision

With the noisy pairing method described above, we can build noisy reviews-summary pairs $\{(\tilde{\mathcal{X}}', y)\}$, but obviously, a model trained with these pairs will generate unfaithful summaries. However, even in such noisy reviews-summary pairs, there would be sub-sequences of the summary y that could be generated from noisy input reviews $\tilde{\mathcal{X}}'$.

To implement this intuition into the training, we first compute the *token-level alignment* between a noisy set of reviews $\tilde{\mathcal{X}}'$ and summary y, and then introduce the indicator function δ_t inside of the standard log-loss function to ignore the unaligned tokens during the training:

$$\mathcal{L}' = -\sum_t \delta_t \log p_\theta(y_t|y_{<t}, \tilde{\mathcal{X}}'),$$

where the alignment function δ_t will be 1 if the token y_t is aligned with the noisy source reviews $\mathcal{X}$ and otherwise 0 as illustrated in Figure 2b. This allows for using aligned words, such as the style and expressions used in the summary, as a training signal without increasing the likelihood of hallucinated words.

For the alignment function, we use word-level matching between the source and target reviews. Since professional writers have a rich vocabulary, which contains words that rarely appear in customer reviews, we implement word stem matching and synonym matching (e.g., serene ∼ calm) to increase the coverage in Partial Supervision. We discuss the design choice of the alignment function in §6.3.

5 Evaluation

We use ProSum and an existing opinion summarization benchmark FewSum (Bražinskas et al., 2020a) to verify the effectiveness and generalizability of Napa🍷. For FewSum, we discarded the source reviews from the training dataset to convert FewSum into a stylized opinion summarization benchmark (i.e., in the non-parallel setting).

5.1 Settings

Training Data For non-parallel training, we first pre-train a self-supervised opinion summarization model using pseudo-reviews-summary pairs (§3). Then, we fine-tune it using noisy reviews-summary pairs using Napa🍷 (§4). Therefore, we need two sets of pseudo-reviews-summary pairs for self-supervised pre-training and noisy reviews-summary pairs for Napa🍷.

As ProSum does not contain customer reviews for training, we use the Yelp review dataset[4], which has 7M reviews for 150k entities, to collect reviews-summary pairs for ProSum dataset. We discarded all the entities used in the Michelin reviews in ProSum to avoid unintentionally selecting the same entity for Noisy Pairing. Then, we excluded entities that do not satisfy the following criteria: (1) in either the `restaurant` or `food` category; (2) the rating is higher than 4.0/5.0 on average. Then, we filtered reviews with 5-star ratings. Finally, we discarded entities that have less than ten reviews. After this pre-processing, we built 100k pseudo-reviews-summary pairs and 1k noisy reviews-summary pairs for self-supervised pre-training and Napa🍷, respectively. The pre-processing method for the FewSum dataset is described in Appendix.

[4] https://www.yelp.com/dataset

Model We instantiate our summarization models using the Transformer model (Vaswani et al., 2017) initialized with the `BART-large` checkpoint (Lewis et al., 2020) in the `transformers` library (Wolf et al., 2020). We used AdamW optimizer (Loshchilov and Hutter, 2019) with a linear scheduler and warmup, whose initial learning rate is set to 1e-5, and label smoothing (Szegedy et al., 2016) with a smoothing factor of 0.1. We tested three configurations: (1) the full version, (2) without Partial Supervision, and (3) without Noisy Paring and Partial Supervision—the self-supervised base model trained only using pseudo-review-summary pairs.

5.2 Baselines

For the main experiment on ProSum, we compared the state-of-the-art opinion summarization system (BiMeanVAE) and two text-style transfer models (Pipeline and Multitask). We also evaluated the upper-bound performance of *Napa*🍷 by using the *parallel* training dataset, where the customer and professionally written reviews for the same entity are correctly paired (Supervised upper-bound). For the FewSum dataset, we compared various opinion summarization models, including self-supervised models and supervised models that use parallel training data, to verify the performance of our non-parallel training framework. The details can be found in Appendix.

BiMeanVAE: BiMeanVAE (Iso et al., 2021) is a self-supervised opinion summarization model based on a variational autoencoder. We further fine-tune this model using Michelin reviews to generate summaries with the desired style.

Pipeline: We combine a self-supervised opinion summarization model and text style transfer model to build a two-stage pipeline. For the self-supervised model, we use the same self-supervised base model as *Napa*🍷. For the text style transfer model, we use STRAP (Krishna et al., 2020), which uses inverse paraphrasing to perform text style transfer using Yelp and Michelin reviews in the non-parallel setting.

Multitask: We use a multi-task learning framework, TitleStylist (Jin et al., 2020), which combines summarization and denoising autoencoder objectives to train a summarization model that generates summaries in the desired writing style. In the experiment, we use Yelp pseudo-reviews-summary pairs (Michelin reviews) for the summarization (denoising) objective.

5.3 Automatic Evaluation

We use the F1 scores of ROUGE-1/2/L (Lin, 2004)[5] and BERTScore (Zhang et al., 2020)[6] for reference-based automatic evaluation. Additionally, we calculate the CTC score (Deng et al., 2021) to evaluate the consistency and relevance of the generated summaries. The consistency score is measured by the alignment between the source reviews and the generated summary based on the contextual embedding similarity; the relevance score is measured by the alignment between the generated summary and the reference summary multiplied by the consistency score. The contextual embeddings are obtained from the `roberta-large` model.

ProSum Table 2 shows the main experimental results on ProSum. The self-supervised model (i.e., *Napa*🍷 w/o Noisy Pairing and Partial Supervision) outperforms all the non-parallel baseline systems. The comparison shows that Pipeline, which combines the self-supervised model and STRAP, degrades the summarization quality. The result indicates that it is not easy to achieve stylized opinion summarization by simply combining a summarization model and a text style transfer model.

Napa🍷 w/o Partial Supervision improves the summarization quality against the self-supervised model while causing degradation in consistency between generated summaries and the source reviews. This degradation is expected, as Noisy Pairing creates pseudo-reviews-summary by sampling reviews from a different entity, only considering the similarity against the pseudo-summary. We will discuss this point in detail in §6.1.

Napa🍷 substantially outperforms the baselines for summarization quality and relevance while maintaining the same level of consistency as the best self-supervised model. This confirms that Partial Supervision successfully alleviates the consistency degradation caused by Noisy Pairing.

The experimental results demonstrate that both Noisy Pairing and Partial Supervision are essential to building a robust stylized opinion summarization model, allowing the model to take advantage of useful signals in the noisy reviews-summary pairs.

FewSum The experimental results on FewSum in the non-parallel setting shown in Table 3 also ob-

[5] `https://github.com/Diego999/py-rouge`
[6] `https://github.com/Tiiiger/bert_score`

	ProSum					
	R1	R2	RL	BS	Consistency	Relevance
Non-parallel baselines						
Multitask (Jin et al., 2020)	23.78	1.85	15.81	80.92	95.01	89.84
Pipeline (Krishna et al., 2020)	27.19	2.69	16.76	82.88	96.69	91.99
BiMeanVAE (Iso et al., 2021)	28.15	3.49	18.68	83.10	96.83	91.98
Napa🍷						
Full version	**33.54**	**4.95**	**20.67**	**84.77**	96.86	**92.48**
w/o Partial Supervision	31.64	3.96	18.90	84.15	96.09	91.80
w/o Noisy Paring and Partial Supervision	28.19	3.43	17.60	83.49	**96.88**	91.92
Supervised upperbound	34.50	5.70	20.64	84.96	97.23	92.96

Table 2: Experimental results on the ProSum dataset. R1/2/L and BS denote the F1 scores of ROUGE-1/2/L and BERTScore. Napa🍷 gives substantial improvements over the baselines. We also confirm that Partial Supervision successfully alleviates the consistency degradation caused by Noisy Pairing.

	Yelp			Amazon		
	R1	R2	RL	R1	R2	RL
Self-supervised baselines						
MeanSum (Chu and Liu, 2019)	27.50	3.54	16.09	26.63	4.89	17.11
CopyCat (Bražinskas et al., 2020b)	28.12	5.89	18.32	27.85	4.77	18.86
Supervised baselines – Parallel training						
FewSum (Bražinskas et al., 2020a)	37.29	9.92	22.76	33.56	7.16	24.49
PASS (Oved and Levy, 2021)	36.91	8.12	23.09	37.43	8.02	23.34
AdaSum (Bražinskas et al., 2022)	38.82	11.75	25.14	39.78	10.80	25.55
BART (our implementation)	39.69	11.63	25.48	39.05	10.08	24.29
Napa🍷 – Non-parallel training						
Full version	**38.59**	**11.23**	**25.29**	**36.21**	**9.18**	**23.60**
w/o Partial Supervision	37.41	10.51	24.18	35.30	7.45	21.92
w/o Noisy Pairing and Partial Supervision	33.39	7.64	20.67	30.18	5.24	19.70

Table 3: Experimental results on the FewSum dataset (Bražinskas et al., 2020a). Napa🍷 shows substantial improvements over the self-supervised baselines. Note that the supervised baseline models were fine-tuned on the parallel training data (i.e., annotated reviews-summary pairs), while Napa🍷 models were trained in the non-parallel setting.

serve the substantial improvements by Napa🍷 over the self-supervised systems. Napa🍷 shows competitive performance against state-of-the-art supervised systems, which use parallel training data for training. The results further confirm that providing a small number of reference summaries in the desired writing style, even if they are not paired with input reviews, can help Napa🍷 train a solid summarization model for stylized opinion summarization.

5.4 Human Evaluation

We conducted human evaluations to compare the performance of our model (Napa🍷) with three baselines: Self-supervision, Pipeline, and Napa🍷 without Partial Supervision (PS) on ProSum with respect to the fluency, relevance, and attractiveness of the generated summary. We asked human annotators recruited from Appen to rate generated summaries on a 4-point Likert scale for each evaluation metric. We describe more details of the human evaluation in Appendix.

Our findings from the results shown in Figure 3 are: (1) using professionally-written summaries for training allows the model to generate more fluent and attractive summaries than other baselines (Napa🍷 and Napa🍷 w/o PS vs. Self-supervision and Pipeline); (2) Napa🍷 without Partial Supervision tends to generate more irrelevant summaries (Napa🍷 vs. Napa🍷 w/o PS). Overall, our results

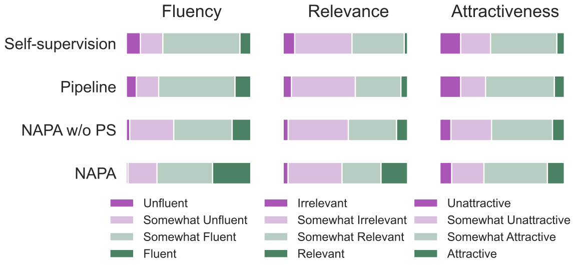


Figure 3: Human evaluations of the fluency, relevance, and attractiveness on ProSum.

demonstrate the importance of using professionally-written summaries for training to improve the fluency and attractiveness of generated summaries and the need for Partial Supervision to ensure the relevance of generated summaries.

6 Analysis

6.1 Importance of Partial Supervision

The experimental results in Tables 2 and 3 show that Napa🍷 without Partial Supervision—just using noisy reviews-summary pairs—demonstrates solid performance for reference-based automatic evaluation metrics. This is a little bit counterintuitive, and this can be attributed to the positive effect of early stopping against noisy training data (Arpit et al., 2017; Li et al., 2020). To analyze this point, we conducted an additional experiment by training Napa🍷 with and without Partial Supervision for more training epochs.

Figure 4 shows the ROUGE-1 F1 score on the validation set of ProSum at different training epochs of the Napa🍷 model trained *with* or *without* Partial Supervision (**orange line** and **green line**). As shown in the figure, we find that in the very early stages of training, both the models improve the ROUGE scores. In the later stage, Napa🍷 *without* Partial Supervision (**green line**) shows continuous degradation, while Napa🍷 *with* Partial Supervision (**orange line**) shows robust performance consistently over the entire training process.

This observation is aligned with the literature on noisy supervision, which shows that overparametrized models learn simple patterns in the early stages of training and then memorize noise (Arpit et al., 2017). On the other hand, it is also known that early stopping is not sufficient under labeling noise (Ishida et al., 2020). We observed that Napa🍷 *without* Partial Supervision generated summaries that were less consistent with the source reviews (Table 2) and contained more hallucinations, as described in Appendix. The results support the importance of Partial Supervision for improving the robustness of the stylized opinion summarization model in non-parallel training.

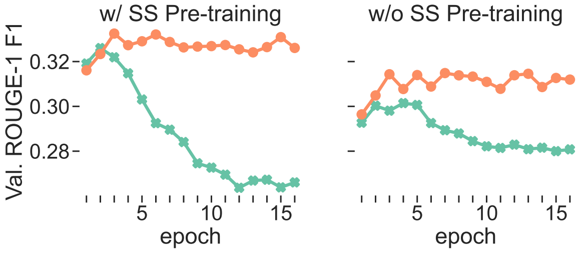


Figure 4: ROUGE-1 F1 score on validation set of ProSum at different training stages. The **orange line** denotes the model trained *with* partial supervision (§4.2), and the **green line** denotes the model trained *without* partial supervision.

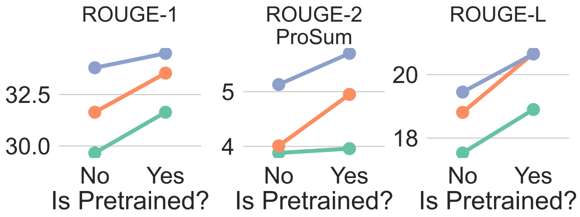


Figure 5: Comparison of summarization quality with and without pre-training. The **blue line** denotes the model trained in a supervised setting, **orange line** denotes the model trained *with* partial supervision and **green line** denotes the model trained *without* partial supervision.

6.2 Pre-training with Self-supervision

As we observe that the self-supervised baseline (i.e., Napa🍷 w/o Noisy Pairing and Partial Supervision) shows solid performance in Table 2 and better performance than the other self-supervised baselines in Table 3, we further investigated the effectiveness of the pre-training using pseudo-reviews-summary pairs (Self-supervision in §3) in the non-parallel training. We conducted ablation studies for the model trained *with* Partial Supervision (**orange line**), *without* Partial Supervision (**green line**), and supervised setting (**blue line**).

As shown in Figure 5, pre-training with self-supervision in all the settings helps improve summarization quality. The effect of pre-training is the most remarkable in the non-parallel settings (**orange line** and **green line**). This indicates that while non-parallel training helps learn the desired writing style for summary generation, it is difficult to determine what content to include in the

	Reference based metrics				Novel n-gram ratios			
	R1	R2	RL	BS	Unigram	Bigram	Trigram	Four-gram
Napa🍷								
No Partial Supervision ($\delta_t = 1$ for all t)	31.64	3.96	18.90	84.15	31.52	80.38	96.54	99.23
+ word match	32.88	4.77	19.98	84.50	12.78	64.10	91.63	97.69
+ word or stem match	32.49	4.82	20.03	84.45	13.23	66.60	92.27	97.94
+ word or stem or synonym match	33.54	4.95	20.67	84.77	15.54	67.19	92.24	97.75
Supervised upperbound	34.50	5.70	20.65	84.96	14.59	58.84	83.20	91.38

Table 4: Comparison of summaries generated with different alignment criteria; + word match is the strictest alignment criterion; adding + stem and + synonym match allows for more relaxed alignment criteria allowing more words to be used for training. As the alignment criteria are relaxed, more novel n-grams can be generated.

summary only from the noisy-reviews-summary pairs. Therefore, we experimentally confirm the effectiveness of self-supervised pre-training for stylized opinion summarization; self-supervision pre-training teaches the model the basics of how to summarize the content, and non-parallel training introduces the model to write in the desired style. The same analysis on the FewSum dataset can be found in Appendix.

6.3 Choice of Token Alignment

As discussed in §4.2, the token alignment function should be carefully chosen to appropriately align customer and professional reviews with different vocabularies. For example, the exact word match should naively disregard semantically similar words (e.g., serene and calm). Thus, we further performed a comparative analysis of the token alignment function. We compared *Napa*🍷 with different variants of Partial Supervision that use: (1) exact word matching, (2) stem matching, and (3) synonym matching.

As shown in Table 4, No Partial Supervision (first row) generates too many novel n-grams, indicating significant hallucinations; it shows the worst summarization performance. We confirm that the model tends to generate more novel n-grams when the alignment criterion is relaxed and also improves summarization performance, suggesting that the stem and synonym matching functions can successfully consider semantically similar tokens to incorporate into training without degradaging the summarization performance.

7 Related Work

Opinion Summarization Due to the challenges in collecting training data, many studies have developed unsupervised solutions for opinion summarization systems (Chu and Liu, 2019; Amplayo and Lapata, 2020; Suhara et al., 2020; Iso et al., 2021; Basu Roy Chowdhury et al., 2022). Recent studies have explored few-shot learning approaches that utilize a small number of review-summary pairs for training (Bražinskas et al., 2020a; Oved and Levy, 2021; Iso et al., 2022).

Our technique falls in the middle of these two approaches, as we do not use annotated reviews-summary pairs for training while using a large number of customer reviews and a small number of professional reviews as auxiliary supervision signals.

Text Style Transfer Text style transfer is a technique to rewrite the input text into the desired style (McDonald and Pustejovsky, 1985). The primary approach for text style transfer is *sentence-level*, which is used as our baselines (Pipeline (Krishna et al., 2020) and Multitask (Jin et al., 2020)).

Based on the observation that both Pipeline and Multitask do not perform well for the stylized opinion summarization task (in Table 2), we confirm that applying sentence-level style transfer cannot offer high-quality stylized opinion summarization and it requires *paragraph-level* text style transfer, which needs further exploration (Jin et al., 2022).

Noisy Supervision Learning statistical models under labeling noise is a classic challenge in machine learning (Angluin and Laird, 1988; Natarajan et al., 2013) and is an active research field because of the increasing availability of noisy data (Han et al., 2020; Song et al., 2022). Among the major approaches for noisy supervision, the loss adjustment approach is widely used in the NLP community, as it can be coupled with any type of commonly used Transformer-based language models (Devlin et al., 2019; Brown et al., 2020)

In text generation, previous studies have attempted to improve the model faithfulness by treating hallucinated summaries as noisy supervision (Kang and Hashimoto, 2020; Fu et al., 2020; Goyal et al., 2022). Our study is different from the line of work in the sense that we combine noisy-reviews-summary pairs and noisy supervision to develop a non-parallel training framework for stylized opinion summarization.

8 Conclusions

This paper proposes stylized opinion summarization, which aims to summarize opinions of input reviews in the desired writing style. As parallel reviews-summary pairs are difficult to obtain, we develop a non-parallel training framework named Noisy Pairing and Partial Supervision (Napa🍷); it creates noisy reviews-summary pairs and then trains a summarization model by focusing on the sub-sequence of the summary that can be reproduced from the input reviews. Experimental results on a newly created benchmark ProSum and an existing opinion summarization benchmark FewSum demonstrate that our non-parallel training framework substantially outperforms self-supervised and text-style transfer baselines while competitively performing well against supervised models that use parallel training data.

9 Limitations

We do not see any ethical issues, but we would like to mention some limitations. This study investigates the use of a limited number of unpaired desired summaries during training. We employ partial supervision to reduce the risk of hallucination, but there is still a potential to generate unfaithful summaries. Thus, the model may generate inconsistent opinions with the source reviews. There is also a trade-off between the quality and diversity of our token-level alignment method. We decided to use exact, stem, and synonym-based matching, but these methods may introduce alignment errors, leading to noisier training. For the annotation tasks, we paid $0.96 for each summary for the crowd workers on Appen. The estimated hourly wage on the platform is $13.48 per hour. For the summary evaluation, we only used token-level matching metrics, unlike LLM-as-a-judge (Liu et al., 2023; Wu et al., 2024).

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Aspect-controllable opinion summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6578–6593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised opinion summarization with noising and denoising. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1934–1945, Online. Association for Computational Linguistics.

Dana Angluin and Philip Laird. 1988. Learning from noisy examples. *Machine Learning*, 2(4):343–370.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR.

Somnath Basu Roy Chowdhury, Chao Zhao, and Snigdha Chaturvedi. 2022. Unsupervised extractive opinion summarization using sparse coding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1209–1225, Dublin, Ireland. Association for Computational Linguistics.

Arthur Bražinskas, 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)*, pages 4119–4135, Online. Association for Computational Linguistics.

Arthur Bražinskas, 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*, pages 5151–5169, Online. Association for Computational Linguistics.

Arthur Bražinskas, Ramesh Nallapati, Mohit Bansal, and Markus Dreyer. 2022. Efficient few-shot fine-tuning for opinion summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1509–1523, Seattle, United States. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Eric Chu and Peter Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR.

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7580–7605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hady Elsahar, Maximin Coavoux, Jos Rozen, and Matthias Gallé. 2021. Self-supervised and controlled multi-document opinion summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1646–1662, Online. Association for Computational Linguistics.

Zihao Fu, Bei Shi, Wai Lam, Lidong Bing, and Zhiyuan Liu. 2020. Partially-aligned data-to-text generation with distant supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9183–9193, Online. Association for Computational Linguistics.

Tanya Goyal, Jiacheng Xu, Junyi Jessy Li, and Greg Durrett. 2022. Training dynamics for text summarization models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2061–2073, Dublin, Ireland. Association for Computational Linguistics.

Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. 2020. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*.

Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. 2020. Do we need zero training loss after achieving zero training error? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4604–4614. PMLR.

Hayate Iso, Xiaolan Wang, Stefanos Angelidis, and Yoshihiko Suhara. 2022. Comparative opinion summarization via collaborative decoding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3307–3324, Dublin, Ireland. Association for Computational Linguistics.

Hayate Iso, Xiaolan Wang, Yoshihiko Suhara, Stefanos Angelidis, and Wang-Chiew Tan. 2021. Convex Aggregation for Opinion Summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3885–3903, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. 2020. Hooks in the headline: Learning to generate headlines with controlled styles. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5082–5093, Online. Association for Computational Linguistics.

Daniel Kang and Tatsunori B. Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. 2020. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4313–4324. PMLR.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

David D. McDonald and James D. Pustejovsky. 1985. A computational theory of prose style for natural language generation. In *Second Conference of the European Chapter of the Association for Computational Linguistics*, Geneva, Switzerland. Association for Computational Linguistics.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Nadav Oved and Ran Levy. 2021. PASS: Perturb-and-select summarizer for product reviews. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 351–365, Online. Association for Computational Linguistics.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. 2020. OpinionDigest: A simple framework for opinion summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5789–5798, Online. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yunshu Wu, Hayate Iso, Pouya Pezeshkpour, Nikita Bhutani, and Estevam Hruschka. 2024. Less is more for long document summary evaluation by LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 330–343, St. Julian's, Malta. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.