Aspect-aware Unsupervised Extractive Opinion Summarization

Haoyuan Li **Somnath Basu Roy Chowdhury** **Snigdha Chaturvedi**
{haoyuanl, somnath, snigdha}@cs.unc.edu
UNC Chapel Hill

Abstract

Extractive opinion summarization extracts sentences from users' reviews to represent the prevalent opinions about a product or service. However, the extracted sentences can be redundant and may miss some important aspects, especially for centroid-based extractive summarization models (Radev et al., 2004). To alleviate these issues, we introduce `TokenCluster` – a method for unsupervised extractive opinion summarization that automatically identifies the aspects described in the review sentences and then extracts sentences based on their aspects. It identifies the underlying aspects of the review sentences using the roots of noun phrases and adjectives appearing in them. Empirical evaluation shows that `TokenCluster` improves aspect coverage in summaries and achieves strong performance on multiple opinion summarization datasets, for both general and aspect-specific summarization. We also perform extensive ablation and human evaluation studies to validate the design choices of our method. The implementation of our work is available at https://github.com/leehaoyuan/TokenCluster.

1 Introduction

In the internet era, online reviews are important for both customers and businesses. Customers use the reviews to help them make better choices while businesses gather feedback from the reviews. However, the large number of reviews for a product can make it time-consuming to go through all of them. Opinion summarization aims to tackle this problem by creating a concise summary of all the reviews. Recently, there has been significant progress in the supervised summarization of single (Liu and Lapata, 2019; Zhong et al., 2020; Liu et al., 2022) and multiple (Fabbri et al., 2019; Pasunuru et al., 2021; Xiao et al., 2022) documents. Unfortunately, opinion summarization cannot directly benefit from the progress as collecting human-written summaries for large-scale review sets is expensive. Therefore, most opinion summarization techniques resort to unsupervised approaches.

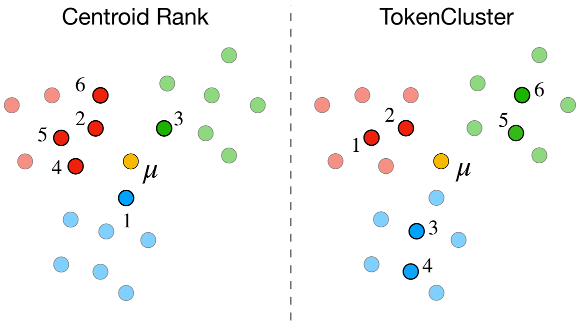


Figure 1: Illustration of a summarization setup using two-dimensional representations (shown as 🔴, 🔵, and 🟢 circles). The color of a circle indicates its *aspect*. (left) Centroid-based summarization techniques greedily select sentences closest to the centroid (μ, shown in 🟡) as the summary (annotated with numbered darker circles). This often results in repetitive, aspect-focused summaries. (right) We present `TokenCluster`, an unsupervised approach for generating summaries that cover diverse aspects, to address this problem.

There are two major paradigms for opinion summarization: abstractive summarization, which generates summaries using novel phrasing, and extractive summarization, where the summary is selected from a subset of the input sentences. For abstractive summarization, previous works usually generate summaries using aggregated sentence representations (Chu and Liu, 2019; Iso et al., 2021a) or train a supervised model on synthetic datasets (Amplayo et al., 2021a; Amplayo and Lapata, 2020). For extractive summarization, recent works use techniques that consists of two primary components: (a) a model for learning representations for review sentences, and (b) an inference algorithm that uses these representations to select summarizing sentences (Angelidis and Lapata, 2018; Angelidis et al., 2021; Chowdhury et al., 2022a). Our work focuses on the inference algorithm for extractive opinion summarization.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12662–12678
July 9-14, 2023 ©2023 Association for Computational Linguistics

Extractive summarization can benefit from aspect identification since extracted sentences should cover all salient aspects within a review set. Aspects in user reviews usually focus on specific features or attributes of an entity (e.g., room, cleanliness, and food for a hotel). Most opinion summarization systems are centroid-based and extract sentences closest to the centroid as the summary. The extraction process can result in redundant summaries that do not cover all aspects. Therefore, it is beneficial to identify aspects of review sentences in order to generate more diverse extractive summaries (as illustrated in Figure 1).

Previous works (Angelidis and Lapata, 2018; Zhao and Chaturvedi, 2020) use aspect seed words to identify salient aspects in reviews. Aspect seed words are sets of words related to the aspect of interest (e.g. 'food', 'restaurant', 'breakfast' for the *food* aspect). These works have shown that using aspect seed words can reduce redundancy and improve the informativeness of summaries. However, extraction of aspect seed words either requires manual annotation (Angelidis and Lapata, 2018) or external data (Zhao and Chaturvedi, 2020). Moreover, aspects for different products or entities may vary based on the domain, requiring additional effort while scaling such techniques to different domains.

Motivated by this observation, we propose `TokenCluster`, an inference algorithm to automatically identify aspects of sentences and extract summarizing sentences based on their aspects. `TokenCluster` identifies the aspect of a sentence using the noun phrases and adjectives that appear in it. After that, it uses a novel inference algorithm for extracting sentences that summarize the general opinion of each aspect. It eventually orders the extracted sentences to improve readability. `TokenCluster` is independent of the underlying sentence representation model and can perform general as well as aspect-specific summarization. Our experiments show that `TokenCluster` outperforms competitive baselines on the Space (Angelidis et al., 2021) and Amazon (Bražinskas et al., 2020) datasets using various sentence representation models. To summarize, our contributions are:

- We propose a novel approach to automatically identify aspects of review sentences.
- We design a novel inference method that extracts summaries covering the most salient aspects.
- `TokenCluster` shows strong performance using both automatic and human evaluation metrics.
- We perform extensive analysis experiments to validate the design of `TokenCluster`.

2 Related Work

Recent opinion summarization methods could be divided into two categories: abstractive and extractive summarization. For abstractive summarization, previous works generate abstractive summaries in two ways. Some of these works (Chu and Liu, 2019; Iso et al., 2021b; Isonuma et al., 2021) use aggregated sentence representations from autoencoders to generate summaries. Others (Bražinskas et al., 2020; Amplayo and Lapata, 2020; Amplayo et al., 2021a,b; Wang and Wan, 2021; Ke et al., 2022) generate synthetic datasets to train generation models in a supervised setting. For extractive summarization, previous works usually follow a two-step approach. In the first step, they learn representations for the review sentences. In the second step (the inference step), they use the learned representations to define relevance scores based on distance from the mean (Chowdhury et al., 2022a,b), distance from the aspect representation (Angelidis et al., 2021) or aspect-specificity and sentiment polarity (Zhao and Chaturvedi, 2020; Angelidis and Lapata, 2018). This work proposes a novel inference algorithm that is independent of the underlying sentence representation model and performs summarization by leveraging the aspects present in a review sentence.

Previous works usually identify aspects in two ways. Some (Angelidis and Lapata, 2018; Zhao and Chaturvedi, 2020; Amplayo et al., 2021a) of them identify aspects using aspect seed words. They generally require manual annotation or external data to obtain aspect seed words. Others (Amplayo et al., 2021b; Wang and Wan, 2021; Ke et al., 2022) use autoencoders to identify the aspect automatically. They do not require human annotation, but they can be noisy since only a portion of words contributes towards the aspect. `TokenCluster` tries to combine the advantages of both approaches. Motivated by Hu and Liu (2004, 2006); Lu et al. (2009), we identify a set of words that are likely to describe the aspect and cluster them into several aspects.

Our work is also related to a contemporary but unpublished work by Bhaskar et al. (2022). Like our method, their method also clusters review sentences based on the distance between a single keyword of each sentence and the aspect seed words. Our method differs from their work in two major aspects. First, in contrast to their approach, TokenCluster extracts multiple aspect-related words from each sentence since a sentence can mention several aspects. Second, TokenCluster does not rely on human-annotated aspect seed words.

3 TokenCluster

In this section, we describe our proposed approach, TokenCluster. We first describe the problem setup (Sec. 3.1). Then, we describe the three components involved of TokenCluster's general summarization algorithm: (a) aspect identification in review sentences (Sec. 3.2), (b) salient sentences identification using aspect information (Sec. 3.3), and (c) sentence ordering to produce the final summaries (Sec. 3.4). Finally, we describe how TokenCluster generates aspect-specific summaries (Sec. 3.5).

3.1 Problem Definition

To perform opinion summarization, we consider an entity e (e.g., a product or hotel) for which a set of reviews are available. Each review contains multiple sentences. We use $S_e = \{s_1, s_2, \ldots\}$ to denote the set of sentences from all reviews for e. The goal is to extract some sentences from S_e in order to summarize the prevalent opinion (general summarization) or to summarize opinions about a particular aspect (aspect-specific summarization).

3.2 Aspect Identification

In this section, we describe how we identify the aspects of review sentences. The aspect identification workflow is shown in Figure 2. To achieve this, we try to locate *aspect-related words* within a sentence, that describe a certain feature of the product or entity being reviewed. Inspired by Hu and Liu (2004, 2006), we consider the roots of noun phrases as aspect-related words since they can directly name the features. For example, for the noun phrase – 'front desk staff', we consider its root – 'staff'. We do not consider pronouns (as it is difficult to perform coreference resolution on user reviews) and proper nouns. We further observe that adjectives can also be indicative of the aspects of sentences. Based on this observation, we incorporate adjectives that do not belong to any noun phrases into the set of aspect-related words. We exclude the adjectives belonging to noun phrases since their aspect information has been covered already by the roots of corresponding noun phrases. We showcase an example of this identification process in Figure 2. Eventually, the set of *aspect-related words* for an entity e, $\mathcal{W}_{\mathsf{asp}}^{(e)} = \{w_1, w_2, ...\}$, includes the roots of noun phrases and adjectives (not belonging to noun phrases) from all sentences in S_e.

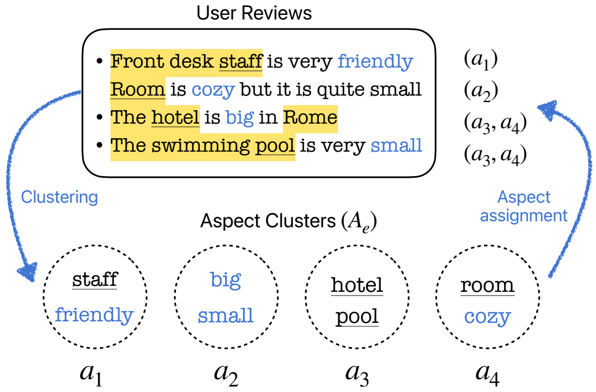


Figure 2: Workflow for aspect identification in user reviews. First, we extract aspect-related words. Noun phrases are highlighted, roots of those noun phrases are underlined, and adjectives (not part of noun phrases) are shown in **blue**. Second, we cluster these aspect-related words to identify latent aspects. Finally, we assign latent aspects to review sentences based on the presence of cluster-specific words.

Next, we obtain the contextual representation for each aspect-related word in $\mathcal{W}_{\mathsf{asp}}^{(e)}$ using a pre-trained BERT$_{\mathsf{base}}$ model (Devlin et al., 2019). We use contextual representations of aspect-related words because words with similar surface forms could describe different aspects. For example, people could use 'nice' to describe that the staff is kind or use it to say 'food' was good. Let $w_i \in \mathcal{W}_{\mathsf{asp}}^{(e)}$ be the k-th word of a sentence $s \in S_e$ containing L words. We feed the sentence s into the pre-trained encoder, $\mathrm{Encoder}(\cdot)$:

$$\mathbf{e} = \mathrm{Encoder}(s) \in \mathbb{R}^{L \times d}, \qquad (1)$$

where d is the representation dimension. We use $\mathbf{w}_i = \mathbf{e}[j]$, the j-th element of $\mathbf{e}$ as the representation for word w_i. Finally, we obtain $\mathbf{W}_{\mathsf{asp}}^{(e)} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots\}$, which denotes the set of representations of words in $\mathcal{W}_{\mathsf{asp}}^{(e)}$.

Despite focusing on noun phrases and adjectives, the set of aspect-related words can contain irrelevant words that do not provide much information about the aspects. To exclude outliers and represent the prevalent aspects, we filter out word representations that are far away from the centroid of word representations $c = \mathbb{E}[\mathbf{W}^{(e)}_{\textsf{asp}}] \in \mathbb{R}^d$. Specifically, we filter out w_i if $\|\mathbf{w}_i - c\|_2$ are ranked among the top γ of $\mathcal{W}^{(e)}_{\textsf{asp}}$.

Next, we want to cluster words that talk about the same aspect together. For example, words like ‘friendly’, ‘nice’, and ‘attentive’ that describe the ‘staff’ aspect should be grouped together. For this, we cluster representations in $\mathbf{W}^{(e)}_{\textsf{asp}}$ into k clusters using Ward hierarchical clustering algorithm (Ward Jr, 1963). We assume that each cluster captures an underlying aspect within the review set. Thereby, an entity has a set of k salient aspects, $A_e = \{a_1, a_2, \ldots, a_k\}$.

So far, we have identified the aspect-related words in the review set. Then, we have clustered these words into k groups that capture underlying aspects. Next, we identify the aspects of a sentence based on the cluster assignment of the aspect-related words within it. In this setup, a review sentence can be assigned multiple aspects. For an entity e, $S^{(e)}_j$ denotes the set of sentences mentioning the aspect a_j.

3.3 Sentence Extraction

Once we assign every review sentence to one or more aspects, we design an algorithm to extract sentences based on their aspect information. For this, we need to obtain a representation $\mathbf{s}_i$, for each sentence, s_i. We can use any sentence representation model to obtain $\mathbf{s}_i$. In our experiments, we use SemAE (Chowdhury et al., 2022a) to retrieve $\mathbf{s}_i \in \mathbb{R}^D$ since it achieves state-of-the-art performance in extractive opinion summarization.

After we retrieve the sentence representations, we calculate the relevance between a sentence and each of the aspects it belongs to (as determined in Sec. 3.2). We define the relevance score $\mathbf{R}(s_i, a_j)$ between sentence s_i, and aspect a_j, which is calculated using KL-divergence:

$$\mathbf{R}(s_i, a_j) = \begin{cases} -\mathrm{KL}(\mathbf{s}_i, \mathbf{a}_j), & \text{if } s_j \in S^{(e)}_j \\ -\infty, & \text{otherwise} \end{cases}, \quad (2)$$

where $\mathbf{a}_j$ is the mean representation of all review sentences containing the j-th aspect, $S^{(e)}_j$:

$$\mathbf{a}_j = \mathbb{E}[\{\mathbf{s}_i | s_i \in S^{(e)}_j\}] \in \mathbb{R}^D. \quad (3)$$

Algorithm 1 Aspect-aware Sentence Extraction

```
1: Input: Aspect set $A_e$, relevance score function $\mathbf{R}(\cdot,\cdot)$, summary budget $B$
2: $\mathcal{U}_{\textsf{asp}} = A_e$    ▷ Initializing uncovered aspects
3: $\mathcal{O}_e = \{\}$    ▷ extracted sentences
4: while $\mathcal{U}_{\textsf{asp}} \neq \emptyset \wedge |\mathcal{O}_e| < B$ do
5:     $r_{\textsf{best}} = -\infty$
6:     for $a \in \mathcal{U}_{\textsf{asp}}$ do
7:         $s_a = \arg\max_s\{\mathbf{R}(s, a) | a \in \textsf{asp}(s)\}$
8:         if $\mathbf{R}(s_a, a) > r_{\textsf{best}}$ then
9:             $s^* = s_a$    ▷ pick $s_a$ with best score
10:            $r_{\textsf{best}} = \mathbf{R}(s_a, a)$
11:        end if
12:    end for
13:    $\mathcal{O}_e = \mathcal{O}_e \cup s^*$    ▷ add to summary
14:    $\mathcal{C}_{\textsf{asp}} = \cup_{s \in \mathcal{O}_e} \textsf{asp}(s^*)$
15:    $\mathcal{U}_{\textsf{asp}} = \mathcal{U}_{\textsf{asp}} \setminus \mathcal{C}_{\textsf{asp}}$    ▷ update $\mathcal{U}_{\textsf{asp}}$
16: end while
17: return $\mathcal{O}_e$
```

A higher $\mathbf{R}(s_i, a_j)$ score indicates that s_i is more representative of the aspect a_j. We use this score to extract summary sentences. However, apart from representing diverse aspects, we want the extracted sentences to be concise and convey as much information as possible within a length budget. To ensure this, we modify the relevance score by introducing a length penalty term to penalize the number of tokens per aspect.

$$\mathbf{R}(s_i, a_j) = \mathbf{R}(s_i, a_j) - \beta \log\left(\frac{|s_i|}{|\textsf{asp}(s_i)|}\right), \quad (4)$$

where β is the weight of length penalty term, $|s_i|$ is the number of tokens in s_i, $\textsf{asp}(s_i)$ is the set of aspects that are covered in s_i. We use the $\log(\cdot)$ function because sentence lengths usually follow a long-tail distribution.

Next, we use the relevance scores to extract sentences that would form the summary. We extract sentences iteratively and ensure the extracted sentences cover all the salient aspects. The extractive summarization routine is described in Algorithm 1. We denote the set of uncovered aspects as $\mathcal{U}_{\textsf{asp}}$ at any point. $\mathcal{U}_{\textsf{asp}}$ is initialized as the entire aspect set A_e (Line 2). At each time step, our goal is to extract one sentence representative of one of the uncovered aspects. For this, we identify one representative sentence, s_a, per uncovered aspect, a, with a corresponding relevance score, $\mathbf{R}(s_a, a)$, and then among all uncovered aspects, extract the sentence with the highest relevance score (to its aspect) for the summary. Specifically, at each time step, we iterate over the uncovered aspects $a \in \mathcal{U}_{\textsf{asp}}$ (Line 6) and identify the sentence s_a

Dataset	Train Ent.	Train Rev.	Rev./Ent.
Space	11.4K	1.14M	100
Amazon	183K	1.46M	8

Table 1: Dataset statistics for Space and Amazon. We report the number of entities in the train set (Train Ent.), the review count in the train set (Train rev.), and the review count per entity in the test set (Rev./Ent.).

with the highest $\mathbf{R}(s, a)$ score (Line 7). Then, we pick the sentence with the highest overall score among all aspects (Line 8, 9, 10) and add it to the summary, $\mathcal{O}_e$ (Line 13). We update the uncovered aspects $\mathcal{U}_{\text{asp}}$ by removing aspects covered by the extracted sentence, s^* (Line 14, 15). We repeat this process until all the aspects are covered or the summary budget (B) is reached (Line 4). Finally, $\mathcal{O}_e$ is the sequence of sentences that form the summary for entity e (Line 17).

3.4 Sentence Ordering

The above-mentioned extraction process does not consider the order of sentences, which might hurt the readability of the summary. For instance, we want sentences mentioning the same aspect to appear together to prevent abrupt context switches between aspects. We also want general sentences to appear first. Ideally, we want the order of extracted sentences to mimic the writing style in human reviews. To achieve these goals, we train a state-of-the-art textual coherence model, ReBart (Basu Roy Chowdhury et al., 2021), on user reviews. We use the trained ReBart to reorder the sentences in $\mathcal{O}_e$. For an entity e, the extractive summary is the concatenation of sentences in $\mathcal{O}_e$ after reordering.

3.5 Aspect-specific Summarization

`TokenCluster` can also perform aspect-specific summarization. In this setup, the user provides a query aspect r, specified by a set of aspect seed words Q_r. We obtain a set of sentences, $S_r^{(e)}$, that contain at least one aspect seed word from Q_r. Then, we obtain the set of aspect-related words for the subset $S_r^{(e)}$ using the aspect identification procedure (described in Sec. 3.2). However, while filtering the aspect-related words, we do not compute the centroid (c) using all aspect-related words ($\mathcal{W}_{\text{asp}}^{(e)}$). We calculate c only using the representations of aspect-seed words:

$$c = \mathbb{E}[\{\mathbf{w}_i | w_i \in Q_r\}] \in \mathbb{R}^d. \qquad (5)$$

This filtering helps reduce the noise and capture aspect-specific information. We follow the same subsequent steps as described in Sec. 3.3 and 3.4.

Method (`Amazon`)	R1	R2	RL
Single			
Random	27.66	4.72	16.95
Centroid$_{\text{BERT}}$	29.94	5.19	17.10
Oracle	31.69	6.47	19.25
Abstractive			
MeanSum (Chu and Liu, 2019)	29.20	4.70	18.15
CopyCat (Bražinskas et al., 2020)	31.97	5.81	20.16
PlanSum (Amplayo et al., 2021b)	32.87	6.12	19.15
TranSum (Wang and Wan, 2021)	34.23	<u>7.24</u>	20.49
COOP (Iso et al., 2021a)	<u>36.57</u>	7.23	<u>21.24</u>
Extractive			
LexRank$_{\text{BERT}}$	31.47	5.07	16.81
QT (Angelidis et al., 2021)	32.08	5.39	16.08
SemAE (Chowdhury et al., 2022a)	32.08	6.03	16.71
`TokenCluster`	**33.40**	**6.71**	**17.95**
w/o length penalty	32.69	6.27	17.68
w/o sentence order	33.40	6.70	17.28

Table 2: General summarization evaluation results on `Amazon` dataset. The best results achieved by extractive systems are shown in **bold**. Overall best results are <u>underlined</u>. We report ROUGE F-scores as – R1: ROUGE-1, R2: ROUGE-2, RL: ROUGE-L.

4 Experimental Setup

In this section, we describe the details of our experimental setup including the datasets, hyperparameters, and baselines.

4.1 Datasets

We perform the experiments on the Space hotel reviews (Angelidis et al., 2021) and the Amazon product reviews (Bražinskas et al., 2020). For the Space dataset, we use the data preprocessed by Angelidis et al. (2021). For the Amazon dataset, we preprocess the data following the instructions of Bražinskas et al. (2020) and exclude the reviews that are not in English using Compact Language Detector 2.[1] The statistics of the datasets are shown in Table 1. The development and test sets of both datasets contain three human-written general summaries per entity. Space dataset also contains three human-written aspect-specific summaries for the following six aspects: building, cleanliness, food, location, rooms, and service.

[1] https://github.com/CLD2Owners/cld2

Method (Space)	R1	R2	RL
Single			
Random	26.24	3.58	14.72
Centroid$_{\text{BERT}}$	31.33	5.78	16.54
Oracle	33.21	8.53	18.02
Abstractive			
MeanSum (Chu and Liu, 2019)	34.95	7.49	19.92
CopyCat (Bražinskas et al., 2020)	36.66	8.87	20.90
AceSum (Amplayo et al., 2021a)	40.37	11.51	23.23
Extractive			
LexRank$_{\text{BERT}}$	31.41	5.05	18.12
AceSum$_{\text{EXT}}$ (Amplayo et al., 2021a)	35.50	7.82	20.09
QT (Angelidis et al., 2021)	37.29	9.12	20.33
SemAE (Chowdhury et al., 2022a)	42.75	12.09	24.84
TokenCluster	44.05	12.81	**26.61**
w/o length penalty	**44.48**	12.81	26.09
w/o sentence order	44.05	**12.84**	26.36

Table 3: General summarization evaluation results on Space dataset. The best results are reported in **bold**. We observe that TokenCluster performs significantly better than the baselines achieving state-of-the-art results.

4.2 Implementation Details

We use spaCy (Honnibal et al., 2020) to identify noun phrases and adjectives. We use BERT$_{\text{base}}$ (Devlin et al., 2019) to get the contextual representations for aspect-related words. Since BERT uses subword tokenization, we average the subwords' representations to get the corresponding word's representation. For ordering the extracted sentences, we trained two versions of ReBART (Basu Roy Chowdhury et al., 2021) on reviews of opinion summarization datasets for 3 epochs.

We train our representation learning model SemAE on Space and Amazon datasets using the same hyper-parameters mentioned in Chowdhury et al. (2022a). The number of clusters, $k = 6$ for the Space and $k = 8$ for Amazon. The aspect-related word filtering percentage, $\gamma = 40\%$ for Space and $\gamma = 20\%$ for Amazon. The weight of length penalty term, $\beta = 5 \times 10^{-3}$ for Space and $\beta = 10^{-2}$ for Amazon. All hyperparameters are tuned on the development set.

4.3 Baselines

We compare our method, TokenCluster, with three types of summarization systems:

Single Review Systems. These systems select a single review as the output summary. We compare TokenCluster with the following: (a) Random randomly samples one review from the set of reviews for an entity, (b) Centroid$_{\text{BERT}}$ selects the review closest to the centroid of all reviews formed using BERT representation, (c) Oracle selects the review with the highest ROUGE score overlap with human summaries.

Extractive Systems. These systems extract sentences from reviews to form summaries. We compare TokenCluster with LexRank (Erkan and Radev, 2004) using BERT representations, QT (Angelidis et al., 2021), Acesum$_{\text{EXT}}$ (Amplayo et al., 2021a), and SemAE (Chowdhury et al., 2022a).

Abstractive Systems. These systems generate summaries using novel phrasing. We compare with MeanSum (Chu and Liu, 2019), Copycat (Bražinskas et al., 2020), PlanSum (Amplayo et al., 2021b), AceSum (Amplayo et al., 2021a), TranSum (Wang and Wan, 2021) and COOP (Iso et al., 2021b).

Method	RL$_{\text{ASP}}$	$\mathbb{E}[N_{\text{ASP}}]$
QT	14.26	4.40
SemAE	15.53	4.52
TokenCluster	15.87	**4.87**
w/o length penalty	16.01	4.67
w/o sentence order	**16.19**	4.87

Table 4: Aspect-awareness evaluation of general summarization on Space dataset. TokenCluster shows better aspect coverage compared to others.

4.4 Results

We measure the performance of summarization models using ROUGE-F1 (Lin, 2004). We report ROUGE-1 (R1), ROUGE-2 (R2), ROUGE-L (RL) for general summarization on the Space dataset in Table 3 and on the Amazon dataset in Table 2. On the Space dataset, TokenCluster achieves state-of-the-art performance on all metrics. On the Amazon dataset, TokenCluster performs the best among all extractive systems and outperforms some abstractive systems like MeanSum, CopyCat, and PlanSum in ROUGE-1 and ROUGE-2. These results show that TokenCluster can effectively select sentences from a large pool of reviews. We attribute the improvement to the enhanced aspect awareness during inference. The improvement over SemAE is statistically significant on all metrics ($p < 0.05$ using paired bootstrap resampling (Koehn, 2004)).

We evaluate the aspect awareness – the degree to which a general summary covers different aspects. We report RL$_{\text{ASP}}$ (Angelidis et al., 2021), which is the average ROUGE-L score using gold aspect-specific summaries as the reference. A higher RL$_{\text{ASP}}$ suggests better aspect coverage. We also report the average aspect coverage,

Method (Space$_{\text{ASP}}$)	Building	Cleanliness	Food	Location	Rooms	Service	$\overline{\text{R1}}$	$\overline{\text{R2}}$	$\overline{\text{RL}}$
Abstractive									
MeanSum	13.25	19.24	13.01	18.41	17.81	20.40	23.24	3.72	17.02
CopyCat	17.10	15.90	14.53	20.31	17.30	20.05	24.95	4.82	17.53
AceSum	-	-	-	-	-	-	32.41	9.47	25.46
Extractive									
LexRank$_{\text{BERT}}$	14.73	25.10	17.56	23.28	18.24	26.01	27.72	7.54	20.82
QT	16.45	25.12	17.79	23.63	21.61	26.07	28.95	8.34	21.77
SemAE	17.82	25.39	**23.64**	**26.76**	24.86	**28.62**	31.78	10.32	24.51
TokenCluster	**22.33**	27.72	22.84	25.61	25.73	27.44	**33.22**	10.86	25.28
w/o length penalty	20.30	27.21	22.55	26.33	**25.95**	27.29	32.89	10.47	24.94
w/o sentence order	21.87	**28.61**	23.35	24.85	25.40	27.99	33.22	**10.89**	**25.35**

Table 5: Evaluation results of aspect-specific summarization on Space dataset. We report the ROUGE-L score for each aspect and the average ROUGE scores over all aspects using $\overline{\text{R1}}$, $\overline{\text{R2}}$ and $\overline{\text{RL}}$. The best results achieved by extractive systems are shown in **bold**. Overall best results are underlined.

Method	Space	Amazon
Reviews	14	12
SemAE	7	13
TokenCluster	9	12
w/o length penalty	11	14

Table 6: Median sentence length of generated summaries. We observe that the length penalty term encourages the model to select concise sentences.

$\mathbb{E}[N_{ASP}]$ (Chowdhury et al., 2022a). N_{ASP} measures the aspect coverage of the generated summaries based on the presence of aspect seed words. A higher $\mathbb{E}[N_{ASP}]$ indicates that the method covers more aspects. The results are reported in Table 4. We observe that TokenCluster can cover more aspects and cover them better.

For aspect-specific summarization, we report the ROUGE-L for each aspect and the average ROUGE-1 ($\overline{\text{R1}}$), ROUGE-2 ($\overline{\text{R2}}$), and ROUGE-L ($\overline{\text{RL}}$) among all aspects. The results are shown in Table 5. TokenCluster achieves the best performance on average ROUGE-1 and ROUGE-2. We observe that the performance gain of TokenCluster over the baselines on aspect summarization is relatively small compared to general summarization. We believe it is because the sentences obtained after filtering using aspect seed words are less diverse. Therefore, the clustering-based extraction strategy becomes less effective. Since the aspect seed words are not very exclusive when querying the specific aspects, some clusters of words might correspond to irrelevant aspects.

4.5 Ablation Study

We investigate the functioning of TokenCluster by ablating several design choices within it.

Length Penalty. We investigate the effect of the length penalty term on the relevance score. For this, we compare TokenCluster with its variant without length penalty term (in Table 2, 3). On Amazon, there is a slight performance drop without the length penalty term. On Space, the ROUGE-1 increases while ROUGE-2 and ROUGE-L drop. In Table 4, there is a drop in $\mathbb{E}[N_{ASP}]$, which suggests that shorter sentences can cover more aspects. We show the median sentence lengths for the two datasets in Table 6. It shows that the length penalty term encourages TokenCluster to select shorter, and hence more concise sentences as summaries.

	Method	R1	R2	RL
Amazon	SemAE	32.08	6.03	16.71
	TokenCluster	**33.40**	**6.70**	**17.28**
	w/ BERT$_{\text{ASP}}$	33.08	6.66	17.42
	w/ BERT$_{\text{all}}$	32.96	6.07	17.00
	w/ SemAE	31.75	5.32	16.40
Space	SemAE	42.75	12.09	24.84
	TokenCluster	**44.05**	**12.84**	**26.36**
	w/ BERT$_{\text{ASP}}$	41.50	11.69	24.45
	w/ BERT$_{\text{all}}$	39.59	10.77	22.93
	w/ SemAE	37.28	8.89	21.39

Table 7: Comparision between TokenCluster using word-level clustering and sentence-level clustering. We observe that TokenCluster with word-level clustering outperforms sentence-level clustering.

Token-level Clustering. To assign aspects for each sentence, we cluster the aspect-related words (see Sec. 3.2). Another possible way of assigning aspects for each sentence is to generate (aspect-related) sentence representations and cluster them. In this section, we compare word-level clustering (TokenCluster's method) with sentence-level clustering. The major difference between these two settings is that a sentence can only contain one aspect when using sentence-level clustering while a

	Method	R1	R2	RL
Amazon	SemAE	32.08	6.03	16.71
	`TokenCluster`	**33.40**	**6.70**	**17.28**
	`w/ all`	32.04	6.17	17.51
	`w/ aspect`	32.32	6.29	17.57
	`w/ NP`$_{\text{root}}$	32.31	5.98	16.86
	`w/ NN+Adj.`	32.81	6.31	17.25
	`w/ NP`$_{\text{root}}$`+Adj.+NNP`	32.40	6.30	17.27
Space	SemAE	42.75	12.09	24.84
	`TokenCluster`	**44.05**	**12.84**	**26.36**
	`w/ all`	41.45	11.94	24.93
	`w/ aspect`	43.37	12.20	24.78
	`w/ NP`$_{\text{root}}$	43.06	12.18	25.33
	`w/ NN+Adj.`	43.45	12.29	25.83
	`w/ NP`$_{\text{root}}$`+Adj.+NNP`	42.70	11.96	25.41

Table 8: Comparison of `TokenCluster` variants using different aspect-related words. We observe that `TokenCluster` using roots of noun phrases and adjectives outperforms the other variants.

`TokenCluster`	
`(w/ sent ordering)`	`(w/o sent ordering)`
This hotel is wonderful. The room was comfortable, the staff was nice and helpful. The room was clean and spacious. The hotel is in a great location and staff were efficient and polite. The restaurant was very good and reasonably priced for a hotel restaurant. The food quality in the restaurant was very good. The front desk staff were also friendly and helpful with any requests.	The room was clean and spacious. The front desk staff were also friendly and helpful with any requests. The hotel is in a great location and staff were efficient and polite. The restaurant was very good and reasonably priced for a hotel restaurant. The room was comfortable, the staff was nice and helpful. This hotel is wonderful. The food quality in the restaurant was very good.

Table 9: Examples of `TokenCluster`'s summaries with or without sentence ordering. We observe that summaries with sentence ordering are easier to read.

sentence can contain multiple aspects using word-level clustering. We consider 3 types of sentence representations for clustering:

- `BERT`$_{\texttt{ASP}}$ generates the aspect-related sentence representation for a sentence by averaging the BERT contextual embeddings of all aspect-related words the sentence contains.
- `BERT`$_{\texttt{all}}$ averages the BERT contextual representations of all words in a sentence to generate the final representation.
- `SemAE` uses representations from SemAE as sentence representations.

We report the results of this experiment in Table 7.[2] We observe that `TokenCluster` outperforms all sentence-level clustering variants by a significant margin except for `BERT`$_{\texttt{ASP}}$ on Amazon, where the performances are comparable. Besides, `BERT`$_{\texttt{ASP}}$ outperforms `BERT`$_{\texttt{all}}$, which suggests that aspect-related words are more important than all words when identifying the aspects of sentences. We also notice that `BERT`$_{\texttt{all}}$ outperforms `SemAE` but not `TokenCluster`. It shows that the improvement is not entirely brought by the BERT model. These results suggest that sentences can contain multiple aspects and explicitly modeling them is more effective for extractive summarization.

Variants of aspect-related words. We consider the roots of noun phrases and adjectives excluding proper nouns as aspect-related words (Sec. 3.2). In this section, we compare this definition with the following five variants: `TokenCluster` (`w/ NP`$_{\text{root}}$) considers only roots of noun phrases excluding adjectives. `TokenCluster` (`w/ NN+Adj.`) includes all nouns and adjectives as aspect-related words without consideration of noun phrases. `TokenCluster` (`w/ NP`$_{\text{root}}$`+Adj.+NNP`) additionally includes proper nouns. `TokenCluster` (`w/ all`) uses all words as aspect-related words. `TokenCluster` (`w/ aspect`) uses the aspect terms extracted by Snippext (Miao et al., 2020), an Aspect-based Sentiment Analysis model. For Amazon dataset, Snippext is finetuned on the laptop domain. For Space dataset, Snippext is finetuned on the hotel domain. results are reported in Table 8. We observe that (`w/ NN+Adj.`) and (`w/ NP`$_{\text{root}}$) outperform SemAE, which shows our method can provide some improvements in summarization with simpler rules of extracting aspect-related words. We also notice that the inclusion of proper nouns leads to a significant performance drop. The potential reason is that it could be difficult for the model to relate the root of proper noun phrases to a certain aspect when the phrases are long and rare. For example, the root of the proper noun phrase, 'Canon 70-200 2.8L' is '2.8L'. It would be difficult to assign this proper noun phrase into the cluster of lenses or cameras. Overall, `TokenCluster` with the original set of aspect-related words outperforms others.

Sentence Ordering. In this section, we explore the effect of sentence ordering on the summary quality in Table 2. We observe that sentence ordering could improve the ROUGE-L score of general summarization on both datasets. However, it leads to a performance drop on RL$_{\text{ASP}}$ and aspect sum-

[2] For simplicity, we evaluate `TokenCluster` without using sentence ordering in this setting.

	Dataset	Inference	R1	R2	RL
SimCSE	Amazon	Mean	32.77	6.33	17.48
		SentCluster	32.17	6.29	17.11
		TokenCluster	**33.03**	**6.35**	**17.53**
	Space	Mean	29.80	4.42	16.57
		SentCluster	38.10	9.47	21.36
		TokenCluster	**40.61**	**10.98**	**23.69**
QT	Amazon	QT	32.08	5.39	16.08
		SentCluster	31.54	5.61	16.86
		TokenCluster	**33.49**	**6.75**	**17.24**
	Space	QT	37.29	9.12	20.33
		SentCluster	32.10	5.14	17.70
		TokenCluster	**40.89**	**10.83**	**24.58**

Table 10: Evaluation results using SimCSE and QT for sentence representation model. TokenCluster achieves the best performance in all setups.

marization. We believe the reason is that we train ReBart on the reviews, which are more similar to general summaries than aspect-specific summaries in terms of content diversity. One example of ordered sentences is shown in Figure 9. We observe that ordering the sentences helps TokenCluster put general sentences at the beginning (shown in green) and avoid abrupt context switches between aspects (shown in red), improving the readability.

Sentence Representation Ablations. As mentioned before, TokenCluster is independent of the sentence representation model. In our implementation, we used SemAE's sentence representation. In this section, we evaluate TokenCluster using SimCSE (Gao et al., 2021) and QT's (Angelidis et al., 2021) sentence representation. For SimCSE, we consider two inference methods as baselines: (i) 'Mean' extracts sentences whose representations are close to the average sentence representation obtained from SimCSE, (ii) 'SentCluster' clusters sentence representations and extracts sentences close to the center of each cluster. For QT, we use the inference method used by Angelidis et al. (2021). When calculating relevance scores (Equation 2), we use Euclidean distance for SimCSE and multi-head cosine similarity for QT. We do not change other hyperparameters and design choices. In Table 10, we observe that TokenCluster significantly outperforms all baseline inference methods except on the Amazon dataset using SimCSE. The results show that TokenCluster can improve extractive summarization performance regardless of the underlying sentence representation models.

Method	Info.	Cohe.	Con.
SemAE	-15.4	-5.3	-38.7
QT	8.0	-16.7	26.7
TokenCluster	7.4	22.0	12.0

Table 11: Human evaluation of general summarization. TokenCluster outperforms SemAE on all three criteria and performs the best in coherence.

4.6 Human Evaluation

We perform the human evaluation on 25 general summaries from the Space dataset using Amazon Mechanical Turk (AMT). We ask the annotators to compare generated summaries from QT, SemAE, and TokenCluster based on three criteria: *informativeness*, *coherence*, and *conciseness*, in a pairwise manner. Each pair of summaries is annotated by three annotators. We compute the scores using Best-Worst Scaling (Louviere et al., 2015). In Table 11, we observe that TokenCluster achieves the best coherence score, and comparable informativeness scores with QT. However, it slightly falls behind QT in terms of conciseness. We observe that QT often generates short repetitive sentences with low aspect coverage, which makes it appear compact. This phenomenon is already captured in our earlier results in Table 4. We report the human evaluation results for aspect-specific summarization in Appendix A.1.

5 Conclusion

We present TokenCluster, a sentence extraction algorithm to automatically identify aspects of sentences and extract sentences based on their aspects. The aspect information helps TokenCluster cover more aspects in summaries. TokenCluster is independent of underlying sentence representation models and shows strong performance on Space and Amazon datasets using multiple sentence representation models. Our extensive analysis shows that our choice of token-based clustering and aspect-related words is optimal compared to its variants. Moreover, using sentence ordering, TokenCluster can generate summaries with better readability. However, TokenCluster can be sensitive to noisy data and may not work well when underlying aspects are not well represented by unique words. Future research can focus on improving the robustness of aspect identification in TokenCluster by incorporating external data and expanding its applicability to diverse domains and languages.

6 Limitations

We propose TokenCluster, a sentence extraction algorithm that can automatically identify aspects in user reviews and leverage that for performing extractive summarization. Being a data-driven approach, it is susceptible to noisy data. Therefore, a limitation of TokenCluster is that the clusters of aspect-related words can be noisy and imbalanced if data is noisy. Future works can focus on more robust ways of extracting and partitioning aspect-related words. This can possibly be achieved using external data. Another limitation is that TokenCluster is computationally more expensive compared to more simple sentence extraction approaches. This is because TokenCluster clusters aspect-related words during inference, which can be restrictive when the review set is large.

7 Ethical Consideration

We do not expect any ethical risks caused by our work. The datasets we use are all publicly available. We do not annotate any data on our own. All our datasets have user reviews in the English language. We performed human evaluation experiments on Amazon Mechanical Turk (AMT). The human annotators were compensated at a rate of $15 per hour. During the evaluation, human annotators were not exposed to any sensitive or explicit content.

8 Acknowledgments

This work was supported in part by NSF grant DRL-2112635. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF. We also thank anonymous reviewers for their constructive reviews.

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021a. 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, Stefanos Angelidis, and Mirella Lapata. 2021b. Unsupervised opinion summarization with content planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12489–12497.

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.

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293.

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*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Somnath Basu Roy Chowdhury, Faeze Brahman, and Snigdha Chaturvedi. 2021. Is everything in order? a simple way to order sentences. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10769–10779, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adithya Bhaskar, Alexander R Fabbri, and Greg Durrett. 2022. Zero-shot opinion summarization with gpt-3. *ArXiv preprint*, abs/2211.15914.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. 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.

Basu Roy Somnath Chowdhury, Chao Zhao, and Snigdha Chaturvedi. 2022a. 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.

Somnath Basu Roy Chowdhury, Nicholas Monath, Avinava Dubey, Amr Ahmed, and Snigdha Chaturvedi. 2022b. Unsupervised opinion summarization using approximate geodesics. *arXiv preprint arXiv:2209.07496*.

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 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR.

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.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. pages 168–177.

Minqing Hu and Bing Liu. 2006. Opinion extraction and summarization on the web. In *Aaai*, volume 7, pages 1621–1624.

Hayate Iso, Xiaolan Wang, Yoshihiko Suhara, Stefanos Angelidis, and Wang-Chiew Tan. 2021a. 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.

Hayate Iso, Xiaolan Wang, Yoshihiko Suhara, Stefanos Angelidis, and Wang-Chiew Tan. 2021b. 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.

Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2021. Unsupervised abstractive opinion summarization by generating sentences with tree-structured topic guidance. *Transactions of the Association for Computational Linguistics*, 9:945–961.

Wenjun Ke, Jinhua Gao, Huawei Shen, and Xueqi Cheng. 2022. Consistsum: Unsupervised opinion summarization with the consistency of aspect, sentiment and semantic. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 467–475.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

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 and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Jordan J Louviere, Terry N Flynn, and Anthony Alfred John Marley. 2015. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.

Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 131–140. ACM.

Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. 2020. Snippext: Semi-supervised opinion mining with augmented data. In *Proceedings of The Web Conference 2020*, pages 617–628.

Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021. Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4768–4779, Online. Association for Computational Linguistics.

Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

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.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Chao Zhao and Snigdha Chaturvedi. 2020. Weakly-supervised opinion summarization by leveraging external information. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9644–9651. AAAI Press.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Method	Aspect Inform.	Aspect Spec.
SemAE	7.8	1.7
QT	-10.0	3.0
TokenCluster	2.2	-4.7

Table 12: Human evaluation results of aspect-specific summarization. We observe that SemAE produces the best aspect-specific summaries.

A Appendix

A.1 Human Evaluation

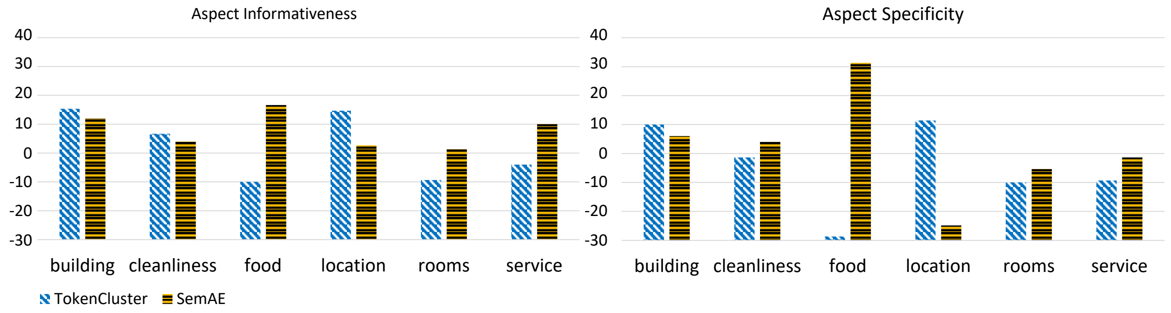
The human annotators are required to be in the United States, have HIT Approval Rate greater than 98, and be masters. The screenshot of the human evaluation interface is shown in Figure 4.

We further perform the human evaluation on 150 aspect-specific summaries in the same manner as human evaluation on general summaries. For aspect-specific summaries, we conduct human evaluation based on two criteria: *aspect informativeness* and *aspect specificity*. The results are shown in Table 12. We observe that TokenCluster does not outperform SemAE and QT on aspect-specific summarization.

Figure 3 shows the best-worst scaling scores of TokenCluster and SemAE per aspect. TokenCluster outperforms SemAE on the location and building aspects but performs poorly on the cleanliness, food, rooms, and service aspects. We compare the aspect-specific summaries of these two models and find two potential reasons for the suboptimal performance on these aspects. First, for the service aspect and the room aspect, most sentences containing corresponding aspect seed words describe the same thing. For example, the retrieved sentences for the service aspect repeat ‘The staff are friendly and helpful’. The extracted sentences of TokenCluster and SemAE are very similar. Therefore, the clustering-based extraction strategy becomes less effective. Second, for the food aspect and the cleanliness aspect, the aspect seed words are too general when querying these aspects. Some clusters of aspect-related words might correspond to irrelevant aspects. For example, ‘good’ is an aspect-seed word for the food aspect. Some sentences containing ‘good’ are not related to the food aspect and might constitute irrelevant clusters during clustering. The same problem also exists for the cleanliness aspect since ‘nice’ is an aspect-seed word for the cleanliness domain. Since clustering-based extraction is less robust to these unrelated sentences compared to centroid-based extraction, the generated summaries of TokenCluster is less specific compared to the generated summaries of SemAE. On the contrary, the sentences retrieved by the aspect-seed words of the building aspect and the location aspect are more diverse. For example, the retrieved sentences of the building aspect can describe a pool, balcony, or lounge. Centroid-based extraction strategy suffers from repetition in this situation. Besides, for these two aspects, the corresponding seed words are more specific and do not include general words like ‘good’ or ‘nice’. Therefore, TokenCluster outperforms SemAE on these two aspects.

A.2 Qualitative Examples

Table 13 shows the clusters of aspect-related words and the summaries generated by TokenCluster. For general summaries, we observe that most clusters corresponds to certain aspects, which helps TokenCluster generate summaries cover more aspects. For the aspect-specific summary, we observe that some clusters are not closely related to the query, which undermines TokenCluster QT’s performance on aspect-specific summarization.

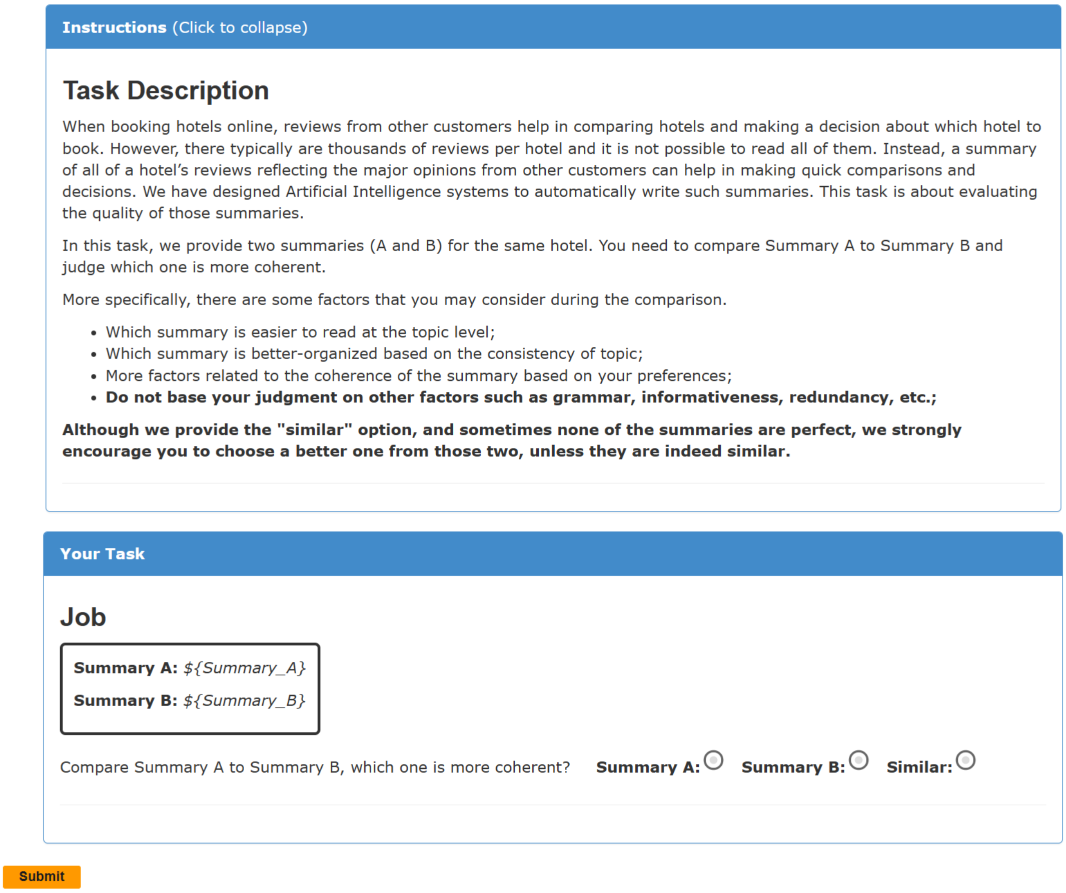

Aspect Informativeness

Aspect Specificity

Figure 3: Best-worst scaling scores of `TokenCluster` and SemAE per aspect. `TokenCluster` performs well on the location and building aspects but performs poorly on the cleanliness, food, rooms, and service aspects.

Space (General)	Amazon (General)	Space (Aspect: Building)
The hotel is very good and clean. The rooms were clean and modern, but quite small. The room was clean and spacious. Breakfast is served in the basement room, with a wide range of food available and I would recommend it. The location was very good and within easy walking distance of everything. The staff were helpful and friendly, and made me feel very welcome. Staff at reception were helpful and friendly.	I got a great deal on this TomTom. The screen is small but for the price, a very nice unit. Sometimes may try to take you on a weird route. I should've known TomTom has a funny way of making you forced to update maps, and in LA a lot of stuff changes so you may as well not even bother with this thing. After researching this problem, it appears there is a known bug that the gps wakes up by itself and drains all battery. very easy to set up and use!	The lobby, restautant, bar and pool looked great at this hotel. However, I still think this hotel is very nice with the size of the room, lobby area, swimming pool and the gym. Beautiful common areas, lounge, lobby, pool.
Cluster 1: location, downstairs, canals. Cluster 2: food, restaurant, cafe. Cluster 3: friendly, pleasant, helpful. Cluster 4: hotel, hotels, room Cluster 5: room, bedrooms, area, unit Cluster 6: staff, guest, manager.	Cluster 1: problem, issue, bug, thing. Cluster 2: maps, route, destination. Cluster 3: lotpleased, help, gift, deal. Cluster 4: thing, system, update. Cluster 5: visuals, features, graphics. Cluster 6: battery, phone, phones.	Cluster 1: area, lounge, loungers, bar. Cluster 2: friendly, restaurant, people. Cluster 3: room, rooms. Cluster 4: pool Cluster 5: lobby, floor, elevators, stairs Cluster 6: hotel, Hilton.

Table 13: Example summaries from `TokenCluster` on Space and Amazon datasets. In the first row, we showcase extracted sentences for the summary. The words of the same color are a cluster of aspect-related words. In the second row, clusters of aspect-related words are shown. The font **colors** indicate the cluster that particular word was part of. We observe that the summarizing sentences cover various clusters (or aspects).



Instructions (Click to collapse)

Task Description

When booking hotels online, reviews from other customers help in comparing hotels and making a decision about which hotel to book. However, there typically are thousands of reviews per hotel and it is not possible to read all of them. Instead, a summary of all of a hotel's reviews reflecting the major opinions from other customers can help in making quick comparisons and decisions. We have designed Artificial Intelligence systems to automatically write such summaries. This task is about evaluating the quality of those summaries.

In this task, we provide two summaries (A and B) for the same hotel. You need to compare Summary A to Summary B and judge which one is more coherent.

More specifically, there are some factors that you may consider during the comparison.

- Which summary is easier to read at the topic level;
- Which summary is better-organized based on the consistency of topic;
- More factors related to the coherence of the summary based on your preferences;
- **Do not base your judgment on other factors such as grammar, informativeness, redundancy, etc.;**

Although we provide the "similar" option, and sometimes none of the summaries are perfect, we strongly encourage you to choose a better one from those two, unless they are indeed similar.

Your Task

Job

Summary A: *${Summary_A}*

Summary B: *${Summary_B}*

Compare Summary A to Summary B, which one is more coherent? **Summary A:** ○ **Summary B:** ○ **Similar:** ○

Figure 4: AMT instructions for human evaluation for general summarization (the figure shows the instructions for coherence measurement). During the evaluation of a certain attribute of a summary (e.g., coherence), we ask the annotators to judge it only based on the attribute and ignore other criteria.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☑ A2. Did you discuss any potential risks of your work?
7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

4.1

☑ B1. Did you cite the creators of artifacts you used?
4.1

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. 4.1, 7

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
4.1

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
4.1,7

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
4.1,7

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4.1

C ☑ Did you run computational experiments?

4

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We use the existing models. Model structure is not the focu of paper

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4.4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix, 4.2

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
4.6

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Appendix

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
7

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
We use public dataset.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. We use public dataset.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
It is not availabel on mturk.