Unsupervised Extractive Opinion Summarization Using Sparse Coding

Somnath Basu Roy Chowdhury **Chao Zhao** **Snigdha Chaturvedi**
{somnath, zhaochao, snigdha}@cs.unc.edu
UNC Chapel Hill

Abstract

Opinion summarization is the task of automatically generating summaries that encapsulate information from multiple user reviews. We present *Semantic Autoencoder* (SemAE) to perform extractive opinion summarization in an unsupervised manner. SemAE uses dictionary learning to implicitly capture semantic information from the review and learns a latent representation of each sentence over semantic units. A semantic unit is supposed to capture an abstract semantic concept. Our extractive summarization algorithm leverages the representations to identify representative opinions among hundreds of reviews. SemAE is also able to perform controllable summarization to generate aspect-specific summaries. We report strong performance on SPACE and AMAZON datasets, and perform experiments to investigate the functioning of our model. Our code is publicly available at https://github.com/brcsomnath/SemAE.

1 Introduction

Opinion summarization is the task of automatically generating digests for an *entity* (e.g. a product, a hotel, a service, etc.), from user opinions in online forums. Automatic opinion summaries enable faster comparison, search, and better consumer feedback understanding (Hu and Liu, 2004; Pang, 2008; Medhat et al., 2014). Although there has been significant progress towards summarization (Rush et al., 2015; Nallapati et al., 2016; Cheng and Lapata, 2016; See et al., 2017; Narayan et al., 2018; Liu et al., 2018), existing approaches rely on human-annotated reference summaries, which are scarce for opinion summarization. For opinion summarization, human annotators need to read hundreds of reviews per entity across different sources for writing a summary, which may not be feasible.

This lack of labeled training data has prompted a series of works to leverage unsupervised or weakly-supervised techniques for opinion summarization (Mei et al., 2007; Titov and McDonald, 2008; Angelidis and Lapata, 2018a; Angelidis et al., 2021). Recent works in this direction have focused on performing opinion summarization in an abstractive setting (Coavoux et al., 2019; Isonuma et al., 2019; Bražinskas et al., 2020; Amplayo et al., 2021b; Iso et al., 2021; Wang and Wan, 2021). Abstractive models are able to produce fluent summaries using novel phrases. However, they suffer from problems common in text generation like hallucination (Rohrbach et al., 2018), text degeneration (Holtzman et al., 2020), and topic drift (Sun et al., 2020). Also, these approaches have been evaluated on small scales (10 reviews per entity or fewer), which does not reveal their utility in the real world where there are hundreds of reviews per entity.

To overcome these issues, another thread of works focuses on extractive opinion summarization, which creates summaries by selecting review sentences to reflect the popular opinions corresponding to an entity. A recently proposed extractive summarization approach is Quantized Transformer (QT) (Angelidis et al., 2021), which leverages vector quantization (van den Oord et al., 2017) for assigning texts to a latent representation that is supposed to capture a semantic sense. However, a text phrase can encapsulate multiple semantic senses, making this representation learning approach restrictive.

Building on the framework introduced by QT, we introduce an unsupervised extractive model, *Semantic Autoencoder* (SemAE), which learns a representation of text over latent semantic units using *dictionary learning* (Dumitrescu and Irofti, 2018). Similar to QT, SemAE leverages Transformer (Vaswani et al., 2017) for sentence reconstruction to simultaneously learn latent semantic units and sentence representations. However, while QT assigns texts to a latent representation (codebook), SemAE models text as a combination of semantics and forms a distribution over latent units (dictionary). This allows sentence rep-

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1209 - 1225
May 22-27, 2022 ©2022 Association for Computational Linguistics

resentations to capture fine-grained and diverse semantics. Unlike QT that relies on identification of aspect-specific head representations, we achieve controllable summarization by utilizing information-theoretic measures (such as relevance, redundancy, etc) on sentence representations. Our sentence selection algorithm is more flexible and allows a broader spectrum of controllable summarization. We experimentally show strong performance on two opinion summarization datasets. Our main contributions are:

- We present Semantic Autoencoder (SemAE), which learns representation of sentences over latent semantic units.
- We introduce novel inference algorithms for general and controllable summarization utilizing information-theoretic measures.
- We show that SemAE outperforms previous methods using automatic and human evaluations.
- We perform analysis to understand how the learnt representations align with human semantics.

2 Related Work

Unsupervised opinion summarization can be conducted either abstractively or extractively. Abstractive approaches aim to summarize the opinion text using novel phrases. Traditional statistical approaches create abstractive summaries using graphical paths (Ganesan et al., 2010) or hand-written templates (Di Fabbrizio et al., 2014). Recent neural approaches leverage the encoder-decoder architecture to aggregate information from multiple reviews and generate summaries accordingly (Chu and Liu, 2019; Bražinskas et al., 2020; Iso et al., 2021; Wang and Wan, 2021).

In contrast to abstractive approaches, extractive approaches rank and select a subset of salient sentences from reviews to form a concise summary (Kim et al., 2011). Saliency computation has been explored using traditional frequency-based approaches (Nenkova and Vanderwende, 2005), similarity with the centroid in the representation space (Radev et al., 2004), and lexical similarity with all sentences in a graph-based representation (Erkan and Radev, 2004). Weakly supervised approaches (Angelidis and Lapata, 2018a; Zhao and Chaturvedi, 2020) extract opinions based on their aspect specificity, and nature of sentiment polarity.

Our work is most similar to the extractive opinion summarization QT (Angelidis et al., 2021) as discussed in Section 1. It is also similar to neural topic model-based approaches (Iyyer et al., 2016; He et al., 2017; Angelidis and Lapata, 2018a) that use a variant of dictionary learning (Elad and Aharon, 2006; Olshausen and Field, 1997) to represent text as a combination of specific semantics (e.g. aspect, relationships etc). In contrast to these models, where text from same topics are trained to have similar representations using max-margin loss, SemAE uses an autoencoder setup to capture diverse latent semantics.

3 Task Description

We follow the task setup in (Angelidis et al., 2021), where given a set of entities (e.g. hotels), a review set $\mathcal{R}_e = \{r_1, r_2, \ldots\}$ is provided for each entity e, where each review r_i is a sequence of sentences $\{s_1, s_2, \ldots\}$. The review set $\mathcal{R}_e$ covers a range of aspects $\mathcal{A} = \{a_1, a_2, \ldots\}$ relating to the domain (e.g. *service*, *location* for hotels). We denote S_e to be the set of sentences from all reviews for an entity e. SemAE is evaluated to perform two types of extractive opinion summarization introduced by Angelidis et al. (2021): (a) *general summarization*, which involves selecting a subset of sentences $O_e \subset S_e$ such that it best represents the reviews in $\mathcal{R}_e$, and (b) *aspect summarization*, where the generated summary $O_e^{(a)} \subset S_e$ focuses on a specific aspect $a \in \mathcal{A}$.

4 The Semantic Autoencoder

The intuition behind Semantic Autoencoder is that instead of representing text as a single latent semantic unit, we represent text as a distribution over latent semantic units using *dictionary learning*. Learning semantic representations over a common dictionary makes them structurally aligned, enabling comparison of sentences using information-theoretic measures.

Semantic Autoencoder consists of three stages (i) *sentence encoding* - an input sentence s is converted into a multi-head representation (H heads) using Transformer encoder $\{\mathrm{s}_h\}_{h=1}^{H}$; (ii) *reconstruction* - a latent representation of head vectors s_h is formed over elements of the dictionary $D \in \mathbb{R}^{K\times d}$, to produce reconstructed representations $\mathbf{z} = \{z_h\}_{h=1}^{H}$; and (iii) *sentence decoding* - a Transformer-based decoder takes as input the reconstructed representations $\mathbf{z}$ to produce the output sentence $\hat{s}$. SemAE is trained on the sentence reconstruction task. The overall workflow of SemAE

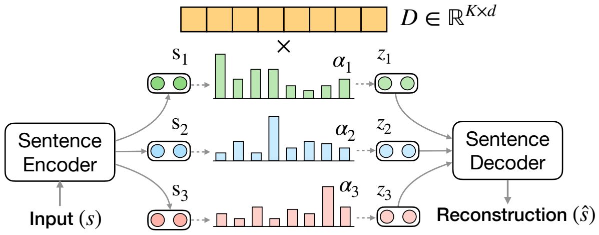

Figure 1: An example workflow of SemAE. The encoder produces $H = 3$ representations (s_h) for a review sentence s, which are used to generate latent representations over dictionary elements. The decoder reconstructs the input sentences using vectors (z_h) formed using latent representations (α_h).

is shown in Figure 1.

4.1 Sentence Encoder

We follow the setup of QT (Angelidis et al., 2021) for sentence encoding. Each sentence s starts with a special token `[SNT]`, which is fed to a Transformer-based encoder. We only consider the final-layer representation of the `[SNT]` token $s_{\text{snt}} \in \mathbb{R}^d$. The sentence representation s_{snt} is split into H contiguous vectors $\{s'_h\}_{h=1}^H$, where $s'_h \in \mathbb{R}^{d/H}$. A multi-head representation is formed by passing s'_h through a layer-normalization layer:

$$\mathrm{s}_h = \mathrm{LN}(s'_h \mathrm{W}^T + b) \tag{1}$$

where $\mathrm{W} \in \mathbb{R}^{d \times d/H}, b \in \mathbb{R}^d$ are trainable parameters and $\mathrm{s}_h \in \mathbb{R}^d$ is the h^{th} head representation.

For each s_h, we obtain a latent representation α_h over the dictionary D, by reconstructing the encoded sentence representation s_h as shown below

$$z_h = \alpha_h D, \quad \alpha_h = \mathrm{softmax}(s_h D^T) \tag{2}$$

where the reconstructed vector $z_h \in \mathbb{R}^d$, and the latent representation $\alpha_h \in \mathbb{R}^K$. We hypothesize that the dictionary D captures the representation of latent semantic units, and α_h captures the degree to which the text encapsulates a certain semantic. The vectors formed $\mathbf{z} = \{z_h\}_{h=1}^H$ are forwarded to the decoder for sentence reconstruction. The dictionary D and s_h are updated simultaneously using backpropagation. For summarization (Section 5), different from QT, we consider α_h (not z_h) as the sentence representation.

4.2 Sentence Decoder

We employ a Transformer-based decoder that takes as input the reconstructed representations $\mathbf{z} = \{z_h\}_{h=1}^H$. MultiHead$(\mathbf{z}, \mathbf{z}, \mathbf{t})$ attention module in the decoder takes $\mathbf{z}$ as key and value, and the target tokens $\mathbf{t}$ as the query. The reconstructed sentence is generated from the decoder as $\hat{s} = \mathrm{Decoder}(\mathbf{z}, \mathbf{t})$. As our goal is sentence reconstruction, we set the target tokens to be same as the input sentence s. Prior work (Angelidis et al., 2021) has also used a similar Transformer-based decoder for sentence reconstruction but they attend directly over quantized head vector formed using codebook elements.

A sentence can capture only a small number of semantic senses. We ensure this by enforcing sparsity constraints on the representations α_h, so that z_h is a combination of only a few semantic units. The encoder, reconstructor and decoder are trained together to minimize the loss function:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(s, \hat{s}) + \lambda_1 \sum_h |\alpha_h| + \lambda_2 \sum_h H(\alpha_h) \tag{3}$$

where $\mathcal{L}_{\text{CE}}$ is the reconstruction cross-entropy loss of the decoder, and to ensure sparsity of α_h we penalize the L1-norm ($|\alpha_h|$) and its entropy $H(\alpha_h)$.

5 Summarization using Latent Representations

We leverage the latent representations α_h generated by SemAE to perform opinion summarization.[1]

5.1 General Summarization

For obtaining the general summary of an entity, we first compute a mean representation of all the review sentences in S_e, which represents the aggregate distribution over semantic units. Thereafter, the general summary is obtained as the collection of sentences that resemble the mean distribution.

Mathematically, every sentence s is associated with a representation over dictionary elements $\alpha^s = [\alpha_1, \ldots, \alpha_H]$, where $\alpha^s \in \mathbb{R}^{H \times K}$. We form the mean representation of review sentences for an entity S_e over dictionary elements as:

$$\bar{\alpha} = \frac{1}{|S_e|} \sum_{s \in S_e} \alpha^s \tag{4}$$

where α^s is the representation for sentence $s \in S_e$.

For general summarization, we compute the *relevance* score $\mathcal{R}(\cdot)$ for each sentence s based on its similarity with the mean representation $\bar{\alpha}$:

$$\mathcal{R}(\alpha^s) = \Delta(\bar{\alpha}, \alpha^s) = -\sum_h \mathrm{KL}(\bar{\alpha}_h, \alpha_h^s) \tag{5}$$

[1] We experimented with different variations of the sentence selection scheme using α_h in Appendix A.4.

where α_h^s is latent representation of sentence s for the h^{th} head. $\Delta(x, y)$ denotes the similarity between two representations x and y. It is implemented as negation of the sum of KL-divergence between head representations. We also experimented with other divergence metrics and observed similar summarization performance (Appendix A.3).

We rank sentences according to descending order of $\mathcal{R}(\cdot)$ and select the top N (a constant hyperparameter, $N < |S_e|$) sentences as the summary O_e (shown in Figure 2). The extracted summary is a concatenation of the text from N selected input sentences (Input (s) in Figure 1). However, modeling relevance only using $\Delta(\cdot, \cdot)$ results in selection of similar sentences. We overcome this by designing variations of our system that have additional information-theoretic constraints.

(a) **Redundancy**: We introduce diversity in the generated summary by penalizing sentences that have a high similarity value with already selected sentences. This is achieved by adding the *redundancy* term in relevance score:

$$\mathcal{R}(\alpha^s, \hat{O}_e) = \Delta(\bar{\alpha}, \alpha^s) - \gamma \max_{s' \in \hat{O}_e} \Delta(\alpha^{s'}, \alpha^s) \quad (6)$$

where $\hat{O}_e$ is the set of sentences selected so far for the summary. The selection routine proceeds in a greedy fashion by choosing $s_0 = \arg\max_{s \in S_e} \Delta(\bar{\alpha}, \alpha^s)$ when $\hat{O}_e = \phi$.

(b) **Aspect-awareness**: Another drawback with sentence selection using $\Delta(\cdot, \cdot)$ is that the summary frequently switches context among different aspects (example shown in Table 7). To mitigate this issue, we identify the aspect of a review sentence using occurrences of aspect-denoting keywords provided in the dataset (Section 5.2). We then cluster the sentences into aspect-specific buckets $\{S_e^{(a_1)}, S_e^{(a_2)}, \ldots\}$ and rank sentences within each bucket. We ignore sentences that are not part of any bucket. We select sentences using two different strategies:

- We iterate over sentence buckets $\{S_e^{(a_i)}\}$ and select the first m sentences ranked according to $\mathcal{R}(\alpha^s)$, from each bucket.

- We prevent selection of similar sentences from a bucket by introducing the redundancy term. We iterate over individual buckets and select first m sentences ranked according to their relevance $\mathcal{R}(\alpha^s, \hat{O}_e^{(a)})$ (Equation 6).

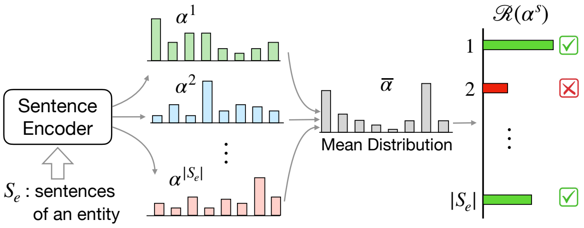

Figure 2: General summary generation routine. The relevance score of each sentence w.r.t mean representation is computed, and top N sentences (O_e) with highest $\mathcal{R}(\cdot)$ are selected as the summary.

5.2 Aspect Summarization

SemAE can perform aspect summarization without needing additional training. For this, we require a small set of keywords to identify sentences that talk about an aspect. For example, *food* aspect is captured using keywords: "breakfast", "buffet" etc.

For a given aspect a, let the keyword set be $Q_a = \{w_1, w_2, \ldots\}$. We use Q_a to identify a set of sentences $S_e^{(a)}$ for each entity e, belonging to aspect a from a held-out dev set S_{dev}. Similar to general summarization, we proceed by computing the mean representation of sentences $S_e^{(a)}$ belonging to the aspect a:

$$\bar{\alpha}^{(a)} = \frac{1}{|S_e^{(a)}|} \sum_{s \in S_e^{(a)}} \alpha_s \quad (7)$$

We then select sentences most similar to the mean representation as the summary.

(a) **Informativeness**: Sentences selected for aspect summarization should talk about the aspect but not the general information. We model *informativeness* (Peyrard, 2019) by ensuring that a selected sentence representation α_s resembles the aspect mean $\bar{\alpha}^{(a)}$, but is divergent from the overall representation mean $\bar{\alpha}$, for a given entity e. For an aspect a, we iterate over sentences in $S_e^{(a)}$ and compute the relevance score for a sentence s as follows:

$$\mathcal{R}_a(\alpha^s) = \Delta(\bar{\alpha}^{(a)}, \alpha^s) - \beta \Delta(\bar{\alpha}, \alpha^s) \quad (8)$$

We rank sentences $s \in S_e$ according to their *aspect-specific relevance* score $\mathcal{R}_a(\cdot)$, and select first N sentences as the summary for aspect $O_e^{(a)}$.[2]

6 Experimental Setup

In this section, we discuss the experimental setup, results and analysis.

[2] We experimented with incorporating the informativeness term in general summarization also but did not find it useful (see Appendix A.3 for more details).

	Reviews	Train / Test Ent.	Rev./Ent.
SPACE	1.14M	11.4K / 50	100
AMAZON	4.75M	183K / 60	8

Table 1: Dataset statistics for SPACE and AMAZON datasets. (Train/Test Ent.: Number of entities in the *training* and *test* set; Rev./Ent.: Number of reviews per entity in the *test* set.)

6.1 Datasets

We evaluated our model on two public customer review datasets SPACE hotel reviews (Angelidis et al., 2021) and AMAZON product reviews (He and McAuley, 2016; Bražinskas et al., 2020). The dataset statistics are reported in Table 1. Test sets of both datasets contain three human-written general summaries per entity. The SPACE corpus was created in a two-step process of sentence selection and then summarization of selected sentences by annotators (further details in Appendix A.2). SPACE dataset also provides human-written summaries for six different aspects of hotels: *building*, *cleanliness*, *food*, *location*, *rooms*, and *service*.

6.2 Implementation details

We build on the implementation framework introduced by Angelidis et al. (2021) for our experiments. We used a 3-layer Transformer with 4 attention heads as the encoder and decoder. The input and hidden dimensions are 320. The encoder and decoder for SemAE was trained for 4 warmup epochs, before the dictionary learning based reconstruction component was introduced. We split the encoded vector into $H = 8$ head representations. We have $K = 1024$ dictionary elements, each with dimension $d = 320$. The dictionary elements are initialized using k-means clustering of review sentence representations. All hyperparameters were tuned on the development set (see Appendix A.1 for more details).

6.3 Metrics

We report ROUGE F-scores that compares the overlap between generated text with gold summaries. For SPACE dataset, we measure how much general summaries cover different aspects by computing the mean ROUGE-L score with the gold aspect summaries (denoted by RL_{ASP}).

We also compute *perplexity* (PPL) score to evaluate the readability of summaries. Perplexity is computed using cross-entropy loss from a BERT-*base* model. We measure *aspect coverage* of a system, by computing the average number of *distinct aspects* N_{ASP} in the generated summaries. Lastly, to evaluate *repetition* in summaries, we compute the percentage of distinct n-grams ($n = 2$).

6.4 Baselines

Following prior work (Angelidis et al., 2021), we compare SemAE with three types of systems:
(a) *Best Review* systems: We report the performance of *Centroid* method, where reviews are encoded using BERT or SentiNeutron (Radford et al., 2017), and the review most similar to the mean representation is selected.
(b) *Abstractive* systems: We report the performance of *Opinosis* (Ganesan et al., 2010) (a graph-based approach), *MeanSum* (Chu and Liu, 2019), *CopyCat* (Bražinskas et al., 2020) and *AceSum* (Amplayo et al., 2021a) summarization models.
(c) *Extractive* systems: We report the performance of *LexRank* (Erkan and Radev, 2004), where sentences were encoded using BERT, SentiNeutron or tf-idf vector. We also report the performance achieved by selecting review sentences *randomly*.

6.5 Results

General Summarization: We present the results of general summarization on SPACE dataset in Table 2. SemAE and its variants show strong improvements over previous state-of-the-art QT, and other baselines, across all ROUGE metrics. They also outperform abstractive systems (like CopyCat and Meansum) by a large margin, which shows that SemAE can effectively select relevant sentences from a large pool of reviews. All variants of SemAE outperform other models in RL_{ASP} metric, showcasing that general summaries from SemAE cover aspects better than baselines. We compiled some baseline results from Angelidis et al. (2021).

We further evaluate the quality of the summaries, for all variations of SemAE along with our strongest baseline QT, using other automatic metrics in Table 3. The first row in Table 3 reports the performance of QT, which achieves the highest distinct n-gram score, but has poor perplexity score. This shows that QT generates summaries with diverse text but they are not coherent. SemAE achieves the best perplexity score (second row in Table 3) but produces less diverse text (lowest distinct n-gram score). The third row in Table 3 reports the performance of SemAE with redundancy term. Comparing rows 2 and 3 of Table 3, we observe that the summaries from SemAE

	SPACE [General]	R1	R2	RL	RL$_{\text{ASP}}$
Best Review	Centroid$_{\text{SENTI}}$	27.36	5.81	15.15	8.77
	Centroid$_{\text{BERT}}$	31.33	5.78	16.54	9.35
	Oracle$_{\text{SENTI}}$	32.14	7.52	17.43	9.29
	Oracle$_{\text{BERT}}$	33.21	8.33	18.02	9.67
Abstract	Opinosis (Ganesan et al.)	28.76	4.57	15.96	11.68
	MeanSum (Chu and Liu)	34.95	7.49	19.92	14.52
	Copycat (Bražinskas et al.)	36.66	8.87	20.90	14.15
	AceSum (Amplayo et al.)	40.37	11.51	23.23	-
Extract	Random	26.24	3.58	14.72	11.53
	LexRank$_{\text{TF-IDF}}$	29.85	5.87	17.56	11.84
	LexRank$_{\text{SENTI}}$	30.56	4.75	17.19	12.11
	LexRank$_{\text{BERT}}$	31.41	5.05	18.12	13.29
	AceSum$_{\text{EXT}}$ (Amplayo et al.)	35.50	7.82	20.09	-
	QT (Angelidis et al.)	38.66	10.22	21.90	14.26
	SemAE	42.48	**13.48**	**26.40**	15.23
	w/ redun.	42.06	12.69	25.77	**15.40**
	w/ aspect	42.86	12.92	25.52	15.22
	w/ aspect + redun.	**43.46**	13.06	25.43	15.14

Table 2: Evaluation results on SPACE dataset. Best results for each metric are shown in **bold**. RL$_{\text{ASP}}$ is the average ROUGE-L score when compared with gold aspect-specific summaries. Systems that access reference summaries are reported in gray.

SPACE [General]	PPL	$\mathbb{E}[N_{\text{ASP}}]$	Distinct-n
QT	4.96	4.40	**0.98**
SemAE	**3.37**	4.44	0.89
w/ redun.	4.01	4.12	0.93
w/ aspect	3.55	**5.24**	0.94
w/ aspect + redun.	3.70	4.84	0.95

Table 3: Evaluation results of QT, SemAE and its different variations on SPACE general summarization. For all setups with redundancy term constant $\gamma = 0.1$.

	AMAZON	R1	R2	RL
Best Rev.	Random	27.66	4.72	16.95
	Centroid$_{\text{BERT}}$	29.94	5.19	17.70
	Oracle$_{\text{BERT}}$	31.69	6.47	19.25
Abstract	Opinosis (Ganesan et al.)	28.42	4.57	15.50
	MeanSum (Chu and Liu)	29.20	4.70	18.15
	CopyCat (Bražinskas et al.)	31.97	5.81	20.16
	PlanSum (Amplayo et al.)	32.87	6.12	19.05
	TranSum (Wang and Wan)	34.23	7.24	20.49
	COOP (Iso et al.)	36.57	7.23	21.24
Extract	LexRank$_{\text{TF-IDF}}$	28.56	3.98	15.29
	LexRank$_{\text{BERT}}$	31.47	5.07	16.81
	QT[†] (Angelidis et al.)	31.27	5.03	16.42
	SemAE	**32.03**	5.38	16.47
	w/ redun.	31.92	**5.68**	**16.61**

Table 4: Evaluation results on AMAZON dataset. Best performance achieved using an extractive systems are in **bold**. Overall best results for each metric is underlined. System performance that access reference summaries are reported in gray.

(w/ redundancy) have more distinct n-grams (less repetition), while falling behind in perplexity and aspect coverage. Performance results for aspect-aware variants of SemAE are reported in last two rows of Table 3. We observe that iteratively covering aspects reduces repetition (increase in distinct-n score). As expected the mean aspect-coverage ($\mathbb{E}[N_{\text{ASP}}]$) improves in aspect-aware SemAE variants. However, a slight drop in aspect-coverage is observed when the redundancy term is introduced (last row in Table 3). We also observe an increase in perplexity for aspect-aware variants, which can be caused due to multiple changes in aspect context. Overall, SemAE (w/ aspect + redundancy) is able to produce diverse text with a high aspect coverage and a decent perplexity score, appearing to be the best performing model.

Evaluation results on AMAZON dataset are reported in Table 4. SemAE and its variants[3] achieve similar performance, with SemAE achieving the best performance among all extractive summarization system. SemAE falls short of only abstractive summarization systems that have the advantage of generating novel phrases not present in the input reviews. Also, while SemAE beats most baselines for AMAZON dataset, the performance gain isn't as much as SPACE dataset. We believe this is because the number of reviews per entity in AMAZON (8) is much lower compared to SPACE (100). As SemAE is dependent on the mean representation $\bar{\alpha}$, having more reviews helps in capturing the popular opinion distribution accurately.[4] For practical purposes, opinion summarization systems are useful when there are hundreds or more reviews per entity. A larger improvement on SPACE shows the efficacy of SemAE in the real world.

Aspect Summarization: For aspect summarization, we compare against four unsupervised systems MeanSum, CopyCat, LexRank and QT on the SPACE dataset. For general summarizers: MeanSum, CopyCat and LexRank, sentence embeddings retrieved from BERT (Vaswani et al., 2017) were clustered using k-means and each cluster $S_e^{(a)}$ was assigned an aspect a based on frequency of aspect-denoting keywords in the cluster's sentences. The models then produced summaries for each aspect a given the input set $S_e^{(a)}$. All models including

[3] We do not have aspect-aware selection variants in AMAZON, as it does not provide aspect-denoting keywords.

[4] We observed a drop in performance when the number of reviews/entity in SPACE dataset was reduced (experimental details in Section 6.6).

[†] Reported results are obtained using the publicly released implementation of QT (Angelidis et al., 2021).

SPACE [Aspect]	Building	Cleanliness	Food	Location	Rooms	Service	$\overline{\text{R1}}$	$\overline{\text{R2}}$	$\overline{\text{RL}}$
MeanSum (Chu and Liu)	13.25	19.24	13.01	18.41	17.81	20.40	23.24	3.72	17.02
CopyCat (Bražinskas et al.)	17.10	15.90	14.53	20.31	17.30	20.05	24.95	4.82	17.53
LexRank$_{\text{BERT}}$ (Erkan and Radev)	14.73	25.10	17.56	23.28	18.24	26.01	27.72	7.54	20.82
QT (Angelidis et al.)	16.45	**25.12**	17.79	23.63	21.61	26.07	28.95	8.34	21.77
SemAE	**20.04**	23.72	**23.57**	**25.33**	**25.29**	**26.90**	**31.24**	**10.43**	**24.14**
w/o informativeness	18.38	24.08	19.03	23.32	23.89	25.05	27.85	8.61	22.29

Table 5: Evaluation results of Aspect Summarization on SPACE dataset. ROUGE-L scores are reported for six different aspects. $\overline{\text{R1}}$, $\overline{\text{R2}}$ and $\overline{\text{RL}}$ are the average ROUGE-1, ROUGE-2 and ROUGE-L F scores respectively. Best system results are in **bold**.

SPACE [General]	Inform.	Coherence	Redund.
QT	-31.3	-47.3	-39.3
SemAE (w/ asp. + redun.)	**-21.3***	**-28.0***	**-27.3***
Human	+52.7	+75.3	+66.7

SPACE [Aspect]	Asp. Inform.	Asp. Specificity
QT	-35.0	-24.7
SemAE	**-13.0***	**-11.0**
Human	+48.0	+35.7

Table 6: Human evaluation results of general and aspect summarization for SPACE dataset. Best human evaluation results obtained for a system are in **bold** and human performance is in gray. (*): statistically significant difference with QT model ($p < 0.05$, using paired bootstrap resampling Koehn (2004)).

SemAE, use the same aspect-denoting keywords.

Evaluation results on SPACE are reported in Table 5. SemAE outperforms the state-of-the-art QT in all aspects except *cleanliness*, where the performance is comparable. We observe that adding the *informativeness* term ($\Delta(\bar{\alpha}, \alpha^s)$ in Equation 8) helps improve the specificity of the aspect thereby boosting performance. SemAE also shows significant gains in terms of average ROUGE-1/2 and ROUGE-L across different aspects.

Human Evaluation: We performed human evaluations for the general and aspect summaries. We evaluated general summaries from QT, best performing variant SemAE (w/ aspect + redundancy) and gold summary. Summaries were judged by 3 human annotators on three criteria: *informativeness*, *coherence* and *non-redundancy*. The judges were presented summaries in a pairwise manner and asked to select which one was better/worse/similar. The scores (-100 to +100) were computed using *Best-Worst Scaling* (Louviere et al., 2015). The first half of Table 6 reports the evaluation results, where we observe that SemAE (w/ aspect + redundancy) outperforms our strongest baseline, QT, for all criteria (statistical significance information provided in the caption of Table 6). However, summaries generated from both systems are far from gold summaries on all criteria.

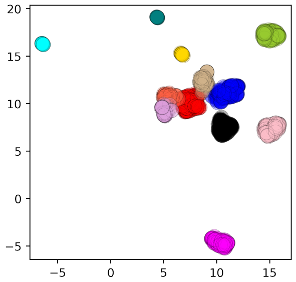

Figure 3: Visualization of UMAP projections of dictionary elements. Projections form clusters, which are shown in different colors.

We also evaluated aspect summaries generated by SemAE and QT in a similar manner. Aspect summaries were judged based on two criteria: *aspect informativeness* (usefulness of opinions for a specific aspect, consistent with reference) and *aspect specificity* (how specific the summary is for an aspect without considering other factors). The bottom half of Table 6 reports the results for aspect summaries. We observe that both QT and SemAE produce aspect-specific summaries. However, SemAE shows a statistically significant improvement over QT in aspect informativeness.

6.6 Analysis

Latent Dictionary Interpretation. In this section, we investigate the semantic meanings learnt by individual dictionary elements, D_k. We visualized the UMAP projection (McInnes et al., 2018) of dictionary element representations (shown in Figure 3). For different runs of SemAE, we found that the dictionary representations converged into clusters as shown in Figure 3 (elements are color-coded according to their cluster identities as assigned by k-means algorithm with k=12).

We hypothesize that the clusters should capture certain semantic meaning. We explore this hypothesis by identifying sentences sharing similar representations with the mean representations

SemAE	**SemAE** (w/ redun.)	**SemAE** (w/ aspect)	**SemAE** (w/ aspect + redun.)
The **staff** is great. The Hotel Erwin is a great **place** to stay. The **staff** were friendly and helpful. The location is perfect. We ate **breakfast** at the hotel and it was great. The hotel itself is in a great **location**. The **service** was wonderful. It was great. The **rooms** are great. The rooftop **bar** HIGH was the icing on the cake. The **food** and **service** at the restaurant was awesome. The **service** was excellent.	The hotel itself is in a great **location**. The **rooms** were clean and we were on the 5th. The best part of the **hotel** is the 7th floor rooftop deck. The **staff** is great. The **hotel** has so many advantages over the other options in the area that it is a no contest. If you want to stay in Venice, this is a great **place** to be. The **food** and **service** at the restaurant was awesome.	The **staff** is great. The **staff** were friendly and helpful. The Hotel Erwin is a great **place** to stay. The **location** is perfect. We ate **breakfast** at the hotel and it was great. The **food** and **service** at the restaurant was awesome. The **rooms** are great. The **room** is epic! The rooftop **bar** HIGH was the icing on the cake. The rooftop **bar** at the hotel, "High", is amazing.	The **staff** is great. We had a great stay at the Erwin, and the **staff** really made it more enjoyable. The Hotel Erwin is a great **place** to stay. It was great. We ate **breakfast** at the hotel and it was great. The **food** and **service** at the restaurant was awesome. The **rooms** are great. We had a **kitchen and balcony** and partial ocean view. The rooftop **bar** HIGH was the icing on the cake.

Table 7: Example summaries from different variants of SemAE. Redundant sentences are highlighted. The aspect denoting words are in **bold**. For SemAE & SemAE (w/ redun.), we observe frequent context switch among aspects. SemAE (w/ aspect) & SemAE (w/ aspect + redun.) summaries cover different aspects in a coherent manner.

(h, k)	Sentences w/ high activation	Explanation
$(3, 5)$	• I wish all hotels or any business for that matter, had employees a dedicated to service as he was. • Very polite and very professional approach.	Service
$(0, 10)$	• Stayed here in August for the our first trip to Vancouver. • I stayed at this motel with my partner in August 2010.	Phrase "stayed"
$(6, 0)$	• Empty water bottles were never thrown out and no one put the iron and ironing board away. • Facing St Paul St can be a very noisy experience.	Bad experience
$(2, 8)$	• A full cooked to order breakfast (including omlettes, . . . , fruit, etc.) • Pizza hut, Mc donalds, KFC all round the corners...	Food
$(5, 8)$	• The rooms seem small, tight fit for a family of 4. • You may have a difficult fit.	Small rooms

Table 8: List of sentences with high activation value with cluster means of dictionary elements. For each head representation, cluster means capture different semantics. h: head index; k: cluster index.

SPACE [General]	5%	10%	50%	100%
Copycat	26.1	26.2	31.8	36.7
QT	36.9	37.1	37.7	38.7
SemAE	37.8	40.9	41.2	42.5

Table 9: ROUGE-1 scores with different training data.

$\{\mu_1, \ldots, \mu_K\}$ for each cluster. For each head h in the encoder (Section 4.1), we compute cosine similarity of sentences with cluster means. Table 8 shows some examples of sentences having highest similarity with a cluster mean μ_k for a head representation h. We observe in most cases sentences closest to a cluster share a similar semantic meaning. For hotel reviews, we observe that sentences often talk about a specific aspect like service, food and rooms, as shown for (h, k) configurations (3, 5), (2, 8) and (5, 8) in Table 8. The clusters sometimes capture certain coarse semantics like presence of a word or phrase (e.g. config. (0, 10) in Table 8). It can also capture high-level semantics like the experience of a customer (e.g. config. (6, 0)). It was interesting to observe that a single cluster can capture different semantics for distinct heads (cluster 8 in configurations (2, 8) and (5, 8)).

Qualitative Examples. Table 7 shows summaries generated by SemAE and its variants for the SPACE dataset. While the summary generated by SemAE talks about *location*, *staff* & *service* multiple times (shown as highlighted text), summary from SemAE (w/ redundancy) doesn't have that repetition.

Also, the summary generated by SemAE switches context frequently. For example, the aspect of the first three sentences changes from service→location→service. We observe that compared to SemAE, both aspect-aware SemAE variants generate summaries without abrupt context switches. The summary generated by SemAE (w/ aspect) covers aspects like service, hotel, food and rooms sequentially, but sentences referring to an aspect are quite similar. SemAE (w/ aspect + redundancy) overcomes this shortcoming, and introduces diversity among the aspect-specific sentences.

Training Data Efficiency. We analyze the performance of SemAE, QT and CopyCat for general summarization (ROUGE-1) on SPACE for varying training data fractions in Table 9. We observe that both QT and SemAE perform well with low training data. However, SemAE outperforms QT in all

Rev./Ent.	R1	R2	RL
5	40.49	12.92	26.23
10	40.76	13.14	26.26
25	41.17	13.18	26.05
50	41.55	13.16	26.01
100	**42.48**	**13.48**	**26.40**

Table 10: ROUGE-F scores of SemAE with varying number of reviews per entity.

low resource settings. SemAE (with 10% data) yields significant ROUGE-1 improvements over QT (with access to 100% data).

Impact of number of reviews. We investigate whether SemAE's performance gain on SPACE is due to the larger number of reviews available (reviews per entity – AMAZON: 8, SPACE: 100). Specifically, we perform ablation experiments by reducing the number of reviews/entity in SPACE dataset. We remove user reviews with low relevance scores (relevance score of a review is the average $\mathcal{R}(\cdot)$ of its sentences). Table 10 reports the performance of SemAE with different number of reviews/entity in the test set. We observe a gradual decline in ROUGE-1 score when the reviews/entity is reduced, which shows that having more reviews per entity helps in better extractive summarization.

Additional Controllable Summarization. We showcase that SemAE can perform different forms of controllable summarization. Specifically, we perform sentiment-based summarization using a small number (10) of seed sentences belonging to *positive*, *negative* and *neutral* sentiment class. Seed sentences were annotated using the rule-based system VADER (Hutto and Gilbert, 2014). An example of sentiment-based summarization is shown in Table 11. We observe SemAE is able to generate summaries aligning with the seed sentiments. We also perform multi-aspect summarization using SemAE, by controlling the aspect of the selected sentences. Table 12 showcases an example of multi-aspect summarization. An interesting observation is that SemAE is able to select sentences, which have mutliple aspects (shown in **blue**) and not independent sentences from different aspects. These experiments show that SemAE is able capture and leverage granular semantics for summarization.

In Appendix A.5, we perform additional analysis to investigate the head-wise analysis, efficacy of sparsity constraints, dictionary evolution, and qualitatively compare SemAE with baselines (QT and CopyCat).

SENTIMENT	SUMMARY
Positive	**Love the warm chocolate chips cookies and the service has always been outstanding.** Excellent morning breakfasts and the airport shuttle runs every 15 minutes but we have made the 10 minute walk numerous times to the airport terminal.
Negative	To add insult to injury, for people who use the parking lot to "park and fly", the charge is $7.95/day, almost half of what the hotel guests are charged!! **Cons - Hotel is spread out so pay attention to how to get to your room as you may get lost**, Feather pillows (synthetic available on request), Pay parking ($16 self/day $20 valet/day), warm cookies on check in.
Neutral	Stayed at this hotel beause the park n fly. **We have stayed at this hotel several times in the family suite (2 bedrooms/1 king and 2 queen beds).** Despite the enormity of this hotel, it very much feels almost family run.

Table 11: An example of sentiment-based summarization for a hotel entity in SPACE dataset.

ASPECTS	SUMMARY
(food, staff)	**The staff was friendly and helpful and we enjoyed the warm, chocolate chip cookie we were given at check-in.** The **breakfast** in the restaurant was amazing, and the **staff** was very attentive.
(room, cleanliness)	**The bed was very nice, room was clean, we even had a balcony.** The beds were comfortable and the room was very **clean**.

Table 12: Examples of multi-aspect summarization for a hotel entity in SPACE dataset.

7 Conclusion

We proposed a novel opinion summarization approach using Semantic Autoencoder, which encodes text as a representation over latent semantic units. We perform extractive summarization by selecting sentences using information-theoretic measures over representations obtained from SemAE. Our experiments reveal that dictionary element representations from SemAE form clusters, which capture distinct semantics. Our model provides fine-grained control to users to model surface-level text attributes (like redundancy, informativeness etc.) in the representation space. SemAE outperforms existing extractive opinion summarization methods on SPACE and AMAZON datasets. Finally, SemAE representations can be leveraged to explore different forms of control on the summary generation (e.g. multi-aspect sumamrization) using our inference framework. Future works can focus on better representation learning systems to handle use-cases with noisy or sparse textual data.

Acknowledgements

This work was supported in part by NSF grants IIS2112635 and IIS2047232. We also thank the anonymous reviewers for their thoughtful and constructive comments.

Ethical Considerations

We do not foresee any ethical concerns from the technology presented in this work. We used publicly available datasets, and do not annotate any data manually. The datasets used have reviews in English language. Human evaluations for summarization were performed on Amazon Mechanical Turks (AMT) platform. Human judges were compensated at a wage rate of $15 per hour.

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.

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. 2018a. 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.

Stefanos Angelidis and Mirella Lapata. 2018b. 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.

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.

Yutian Chen, Max Welling, and Alexander J. Smola. 2010. Super-samples from kernel herding. In UAI 2010, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, July 8-11, 2010, pages 109–116. AUAI Press.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 484–494, Berlin, Germany. Association for Computational Linguistics.

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

Maximin Coavoux, Hady Elsahar, and Matthias Gallé. 2019. Unsupervised aspect-based multi-document abstractive summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Pierre Colombo, Guillaume Staerman, Chloé Clavel, and Pablo Piantanida. 2021. Automatic text evaluation through the lens of Wasserstein barycenters. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10450–10466, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Giuseppe Di Fabbrizio, Amanda Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. In Proceedings of the 8th International Natural Language Generation Conference (INLG), pages 54–63, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.

Bogdan Dumitrescu and Paul Irofti. 2018. Dictionary learning algorithms and applications. springer.

Michael Elad and Michal Aharon. 2006. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image processing, 15(12):3736–3745.

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.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages

340–348, Beijing, China. Coling 2010 Organizing Committee.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 388–397, Vancouver, Canada. Association for Computational Linguistics.

Ruining He and Julian J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016, pages 507–517. ACM.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 168–177.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the International AAAI Conference on Web and Social Media, volume 8.

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.

Masaru Isonuma, Junichiro Mori, and Ichiro Sakata. 2019. Unsupervised neural single-document summarization of reviews via learning latent discourse structure and its ranking. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2142–2152, Florence, Italy. Association for Computational Linguistics.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former Friends: Unsupervised learning for dynamic fictional relationships. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1534–1544, San Diego, California. Association for Computational Linguistics.

Hyun Duk Kim, Kavita Ganesan, Parikshit Sondhi, and ChengXiang Zhai. 2011. Comprehensive review of opinion summarization.

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.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net.

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

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. ArXiv preprint, abs/1802.03426.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 5(4):1093–1113.

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007, pages 171–180. ACM.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehree, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005, 101.

Bruno A Olshausen and David J Field. 1997. Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision research, 37(23):3311–3325.

Bo Pang. 2008. lee, l.(2008). opinion mining and sentiment analysis. Foundations and trends in information retrieval, 2(1-2):1–135.

Maxime Peyrard. 2019. A simple theoretical model of importance for summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1059–1073, Florence, Italy. Association for Computational Linguistics.

Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning, 11(5-6):355–607.

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

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. ArXiv preprint, abs/1704.01444.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

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

Ruixiao Sun, Jie Yang, and Mehrdad Yousefzadeh. 2020. Improving language generation with sentence coherence objective. ArXiv preprint, abs/2009.06358.

Ivan Titov and Ryan T. McDonald. 2008. Modeling online reviews with multi-grain topic models. In Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008, pages 111–120. ACM.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 6306–6315.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

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.

Chao Zhao and Snigdha Chaturvedi. 2020. Weakly-supervised opinion summarization by leveraging external information. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 9644–9651.

DATASET	λ_1	λ_2
AMAZON	10^3	5×10^{-4}
SPACE	10^4	5×10^{-4}

Table 13: Loss function hyperparameters values.

A Appendix

A.1 Implementation Details

The Transformer is trained without the dictionary learning reconstruction for 4 warmup epochs. We tokenized text in an unsupervised manner using SentencePiece[5] tokenizer with 32K vocabulary size. The model was trained using Adam Optimizer with a learning rate of 10^{-3}, and a weight decay of 0.9. Our model was trained for 10 epochs on a single GeForce GTX 2080 Ti GPU in 35 hours. The loss function parameters are reported in Table 13. The hyperparameters were tuned on the development set of the dataset based on ROUGE-1 F score. For aspect summarization, we set $\beta = 0.7$ after tuning (grid search between 0.1 and 1, with intervals of 0.1) on the development set. We choose the redundancy term constant $\gamma = 0.1$ in a similar manner. Post training, the summaries were generated with $N = 20$. We limit the summary length to 75 tokens. Each keyword $w_i \in Q_a$ is associated with a confidence score for aspect a. In case a sentence has multiple keywords belonging to different aspects we use the confidence score to assign the aspect.

A.2 Dataset Construction

In this section, we provide some background information about the dataset creation process for SPACE and AMAZON. SPACE corpus has a large number of reviews per entity. Therefore, Angelidis et al. (2021) collected summaries from reviews following a two-step procedure (a) sentence voting, and (b) summary collection. *Sentence voting* step involves selecting informative review sentences using a majority vote from the annotators. Annotators were prompted to select between 20-40% of the total sentences. *Summary collection* involves generating a overview summary of the selected sentences upto a 100-word budget. For aspect summaries, selected sentences were annotated using an off-the-shelf aspect classifier (Angelidis and Lapata, 2018b). Human annotators were asked to summarize selected sentences belonging to an aspect. AMAZON dataset has a relatively lower number of reviews per entity. The evaluation set of AMAZON was created by sampling 60 entities and 8 reviews per entity. These were provided to the human annotators for summarization (Bražinskas et al., 2020).

SPACE	R1	R2	RL	PPL	$\mathbb{E}(N_{ASP})$	Dist. n
SemAE	42.48	13.48	26.40	3.37	4.44	0.89
w/ cosine Δ	42.53	13.67	26.12	3.41	4.44	0.89
w/ inform.	42.48	13.47	26.13	3.32	4.44	0.89

Table 14: Evaluation results of ablation experiments. For informativeness term, $\beta' = 0.1$.

A.3 Ablations

- **Divergence metric**: SemAE uses KL divergence to measure the relevance of a sentence α^s when compared to the mean $\bar{\alpha}$, we used KL-divergence earlier. In this setup, we experiment with cosine similarity as our divergence function $\Delta(\cdot,\cdot)$. The modified divergence $\Delta(\cdot,\cdot)$ score is defined as:

$$\Delta(\alpha^s, \bar{\alpha}) = \sum_h \frac{\bar{\alpha}_h^T \alpha_h^s}{||\bar{\alpha}_h||_2 ||\alpha_h^s||_2} \quad (9)$$

  The second row in Table 14 reports the performance in this setup, which is similar to the baseline SemAE performance. This shows that cosine similarity can serve as a good proxy to measure relevance $\mathcal{R}(\cdot)$.

- **Informativeness**: In this ablation experiment, we incorporate the informativeness term in general summarization. The modified relevance score is:

$$\mathcal{R}(\alpha^s) = \Delta(\bar{\alpha}, \alpha^s) - \beta' \Delta(\alpha^{(b)}, \alpha^s) \quad (10)$$

  where $\alpha^{(b)} = \mathbb{E}[\alpha^s]$, the mean representation of all sentences across all entities. $\alpha^{(b)}$ captures *background knowledge* distribution (Peyrard, 2019), and a good summary should be divergent from the background information. Third row in Table 14 reports the performance in this setup, where we do not observe any gain over the baseline. We believe this maybe due to the fact that $\alpha^{(b)}$ doesn't capture the background knowledge properly, as it is the mean representation of hotel review sentences across all entities.

  For both ablation setups, we observe almost no change in perplexity, aspect coverage and distinct n-grams metrics.

[5] https://github.com/google/sentencepiece

DATASET	METHOD	R1	R2	RL
SPACE	SemAE	**42.48**	**13.48**	**26.40**
	w/ Herding	39.69	10.30	22.81
	w/ Optimal Transport	38.38	9.34	22.38
	w/ Clustering	30.00	4.35	17.66
AMAZON	SemAE	**32.03**	**5.38**	16.47
	w/ Herding	30.36	4.95	15.67
	w/ Optimal Transport	31.45	5.23	**17.12**
	w/ Clustering	31.42	5.27	16.58

Table 15: Summarization performance of SemAE with different sentence selection schemes on SPACE and AMAZON datasets.

A.4 Variations of Sentence Selection

(a) **Herding** (Chen et al., 2010): In this setup, we modify selection mechanism of SemAE by updating the mean representation every time a sentence is selected. We consider the mean of the sentences that have not been selected so far. The intuition behind this approach is that the next selected sentence should best capture information, which is not present in the summary so far. The sentence selection process is described below:

$$\alpha_t^s = \max_{\alpha^s} \mathcal{R}(\alpha^s) = \max_{\alpha^s} \Delta(\bar{\alpha}_t, \alpha^s) \qquad (11)$$

$$\bar{\alpha}_t = \mathop{\mathbb{E}}_{s\sim(S_e\backslash\hat{O}_e)} [\alpha^s] \qquad (12)$$

where α_t^s is the representation selected at time step t, $\bar{\alpha}_t$ is mean representation of the set of sentences that are not part of the summary yet and $\hat{O}_e$ is the set of selected sentences so far. Table 15 reports the result of this setup. We observe a significant drop in performance compared to SemAE. We believe that removing the selected sentences skews the mean towards outlier review sentences resulting in a drop in performance.

(b) **Optimal Transport**: In this setup, we consider the Wasserstein distance between two probability distributions. Wasserstein distance (Peyré et al., 2019) arising from the concept of optimal transport takes into account the underlying geometry of the representation space. Let $\mathcal{M}_+^1(\mathbb{R}^d)$ be the space of probability distributions defined on $\mathbb{R}^d$ with $d \in \mathbb{Z}^+$. Wasserstein distance between two arbitrary probability distributions $\mu \in \mathcal{M}_+^1(\mathcal{X})$ and $\nu \in \mathcal{M}_+^1(\mathcal{Y})$ is denoted by $\mathcal{W}(\mu, \nu)$. Following (Colombo et al., 2021), we compute a Wasserstein barycenter of all sentences for each head h as:

$$\mu_h^c = \mathop{\arg\min}_{\mu\in\mathcal{M}_+^1(\mathbb{R}^d)} \sum_{i=1}^{|S_e|} \mathcal{W}(\mu, \alpha_h^s) \qquad (13)$$

The overall representation for the barycenter is $\mu^c = [\mu_1^c, \ldots, \mu_H^c]$. Next, we derive the relevance score of each sentence s with the barycenter as:

$$\mathcal{R}(\alpha^s) = -\sum_{h=1}^{H} \mathcal{W}(\mu_h^c, \alpha_h^s) \qquad (14)$$

As shown in Equation 14, we select sentences with low Wasserstein distance from the barycenter. We report the results for this optimal transport setup in Table 15. We find that the performance of this setup is significantly lower than SemAE on SPACE dataset, but comparable to other baselines on AMAZON dataset.

(c) **Clustering-based Sentence Selection**: In this setup, instead of selecting sentences similar to the mean representation, we identify clusters formed by the representations. For clustering we flatten the sentence representation $\alpha^s \in \mathbb{R}^{HK}$, and use k-means[6] clustering (K is a hyperparameter). We select sentences that are representative samples in each cluster. The relevance score for each sentence is computed as follows:

$$\mathcal{R}(\alpha^s) = -||\alpha^s - \alpha_{\mathcal{C}}||_2^2 + \gamma|\mathcal{C}| \qquad (15)$$

where $\alpha_{\mathcal{C}}$ is the representation of the cluster center where s belongs, and $|\mathcal{C}|$ is the size of the cluster. The first term in Equation 15 penalizes the relevance of a sentence for being too far away from the cluster center, and the second term selection of samples from a large cluster. The hyperparameters $\gamma = 0.005, K = 5$ in our experiments, were selected using the development set performance. In Table 15, we observe that this clustering-based sentence selection work poorly for SPACE dataset but the performance on AMAZON is decent. The performance on SPACE dataset is poor as it has a large number of reviews, and identification of representative clusters is difficult using this approach.

A.5 Extended Analysis

(a) **Efficacy of Sparsity Losses**: In this section, we evaluate the performance of SemAE in different

[6] We experimented with algorithms (like Affinity Propagation, DBSCAN) that identify clusters automatically, but found them to struggle with outliers. K-means performed better than them albeit requiring finetuning of the hyperparameter.

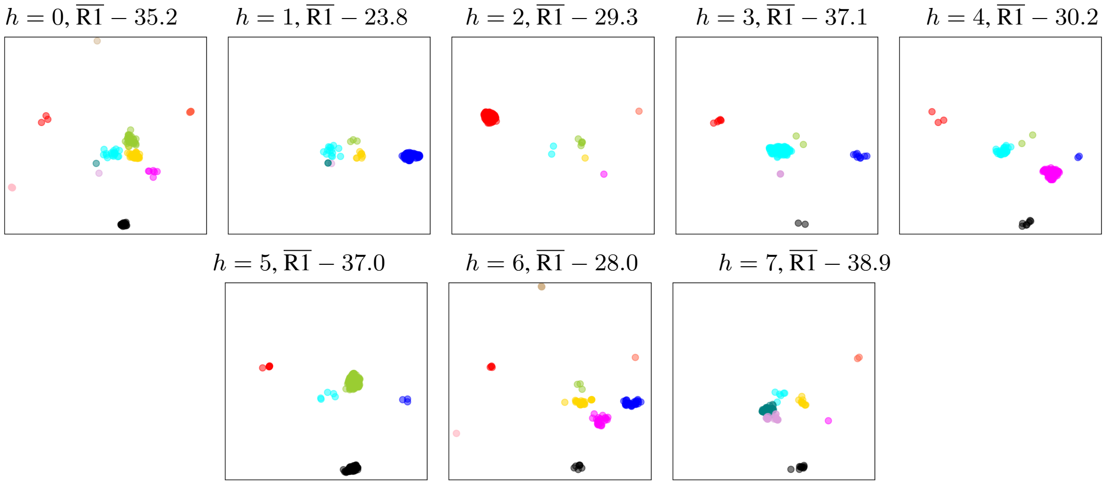

Figure 4: Head-wise visualization of UMAP (McInnes et al., 2018) dictionary element projections.

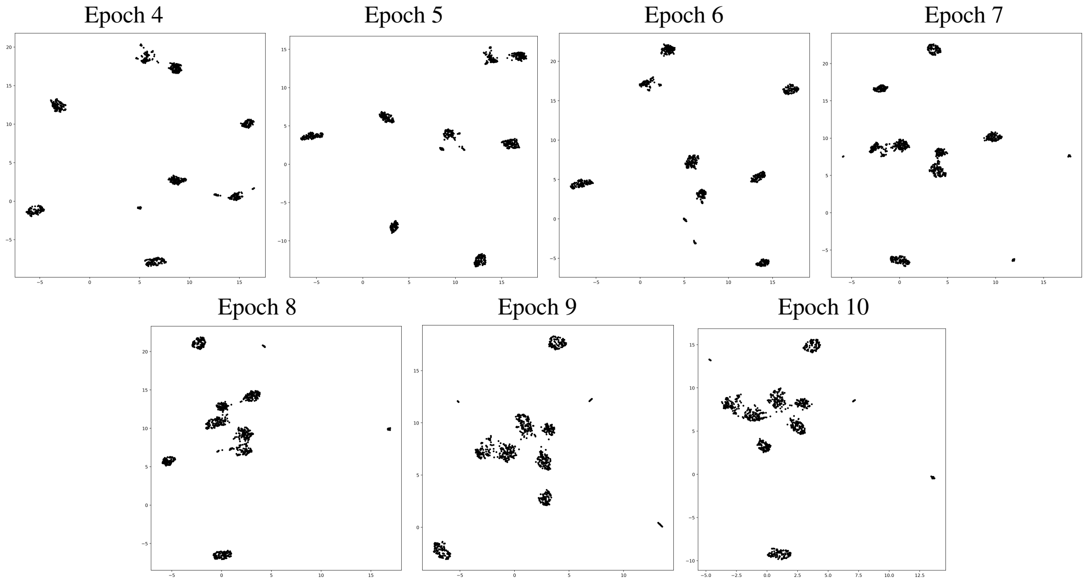

Figure 5: UMAP (McInnes et al., 2018) projections of dictionary element over different epochs (warmup epoch #4 to epoch 10). We observe that dictionary elements gradually evolve to form clusters over the epochs.

DATASET	METHOD	R1	R2	RL
SPACE	SemAE	42.48	13.48	26.40
	w/o L1	41.01	11.91	24.23
	w/o H	38.70	10.45	22.87
AMAZON	SemAE	32.03	5.38	16.47
	w/o L1	29.16	4.77	16.19
	w/o H	29.60	4.85	16.63

Table 16: Performance of SemAE in different configurations of sparsity constraints.

configurations of sparsity losses. Specifically, we gauge SemAE' performance when L1-loss and entropy loss are removed. Table 16 reports the results with different loss setups. We observe a drop in performance when either of the sparsity losses are removed. This shows that ensuring sentence representations are a sparse combination of semantic units helps in summarization.

(b) **Head-wise Analysis**: We analyze whether there is a correlation between the head-wise representations and clusters formed by dictionary elements. For each dictionary element, we compute the average attention (α_h) it receives from each head h, and assign the element to a head where it received the maximum mean attention (head-wise dictionary elements are shown in Figure 4). We also compute the performance of general summarization when only a single head representation is considered $\Delta(\alpha^s, \bar{\alpha}) = \text{KL}(\bar{\alpha}_h, \alpha_h^s)$. In Figure 4, we observe that heads that have instances in mul-

Human	SemAE	QT	Copycat
All staff members were friendly, accommodating, and helpful. The hotel and room were very clean. The room had modern charm and was nicely remodeled. The beds are extremely comfortable. The rooms are quite with wonderful beach views. The food at Hash, the restaurant in lobby, was fabulous. The location is great, very close to the beach. It's a longish walk to Santa Monica. The price is very affordable.	The staff is great. The Hotel Erwin is a great place to stay. The staff were friendly and helpful. The location is perfect. We ate breakfast at the hotel and it was great. The hotel itself is in a great location. The service was wonderful. It was great. The rooms are great. The rooftop bar HIGH was the icing on the cake. The food and service at the restaurant was awesome. The service was excellent.	Great hotel. We liked our room with an ocean view. The staff were friendly and helpful. There was no balcony. The location is perfect. Our room was very quiet. I would definitely stay here again. You're one block from the beach. So it must be good! Filthy hallways. Unvacuumed room. Pricy, but well worth it.	This hotel is in a great location, just off the beach. The staff was very friendly and helpful. We had a room with a view of the beach and ocean. The only problem was that our room was on the 4th floor with a view of the ocean. If you are looking for a nice place to sleep then this is the place for you.

Table 17: Human-written and system generated summaries from SemAE, QT and Copycat. We showcase the summary for the same instance reported by previous works.

Food: The food and service at the restaurant was awesome. The food at Hash, the restaurant just off of the lobby, was fabulous for breakfast. The food was excellent (oatmeal, great wheat toast, freshberries and a tasty corned beef hash).	
Location: The Hotel Erwin is a great place to stay. The hotel is not only in the perfect location for the ideal LA beach experience, but it is extremely hip and comfortable at the same time.	
Cleanliness: The room was spacious and had really cool furnishings, and the beds were comfortable. The room itself was very spacious and had a comfortable bed. We were upgraded to a partial ocean view suite and the room was clean and comfortable.	
Service: The hotel staff were friendly and provided us with great service. The staff were friendly and helpful. The staff was extremely helpful and friendly. The hotel staff was friendly and the room was well kept.	
Building: The rooftop bar at the hotel, "High", is amazing. The rooftop bar HIGH was the icing on the cake. The Hotel Erwin is a great place to stay. The best part of the hotel is the 7th floor rooftop deck.	
Rooms: The room was spacious and had really cool furnishings, and the beds were comfortable. The room itself had a retro 70's feel with a comfortable living room and kitchen area, a separate bedroom with a nice king size bed, and a sink area outside the shower/toilet area.	

Table 18: Aspect-wise summaries generated by SemAE.

tiple dictionary element clusters ($h = 0, 3, 5, 7$) perform better than heads where instances are concentrated over few clusters ($h = 1, 2$).

(c) **Output summaries**: Table 17 shows the summaries generated by SemAE, QT and Copycat along with human-written summary. We observe that SemAE selects well formed sentences, avoiding truncated sentences or the ones in a first-person setting. Table 18 reports the summaries generated by SemAE for different aspects of a hotel entity. We observe that SemAE is able to produce summaries that talk about the specific aspect only.

(d) **Evolution of Dictionary Representations**: We plot the UMAP projections of dictionary elements from epochs 4 (after encoder warmup is complete) to 10 in Figure 5. During the training process, we observe that the UMAP project of dictionary elements form a set of clusters. We observe the first signs of cluster formation in epoch 7, which becomes more distinct over the later epochs.

SPACE [General]	R1	R2	RL
QT	36.1	7.6	20.2
QT (+SS)	35.7	8.1	22.4
SemAE	**37.8**	**9.7**	**22.8**

Table 19: Summarization performance with SemAE's sentence selection (SS) scheme using representations from QT and SemAE. We also report the performance of the baseline QT. The experiments were conducted on 5% SPACE dataset.

(e) **Ablations with QT**: In this section, we analyze the efficacy of our sentence selection (SS) module. We evaluate the summarization performance using our sentence selection scheme by retrieving sentence representations from QT and SemAE. The experiments were performed using 5% data from the SPACE dataset. For QT's representations, we obtain α_h (Equation 2) as follows:

$$\alpha_h = \mathrm{softmax}(-||s_h - D||_2^2) \tag{16}$$

In Table 19, we observe that incorporating our sentence selection (SS) improves QT's performance in terms of ROUGE-2 and ROUGE-L scores, with a small drop in ROUGE-1. However, the performance still falls behind SemAE, showcasing that the our representation learning model complements the sentence selection scheme. From these two results, we can conclude that the better performance of SemAE can be attributed to a combination of the two components. Note that using QT's sentence selection with SemAE's representations is not feasible as SemAE doesn't quantize sentences to a single latent code.