Extractive Opinion Summarization in Quantized Transformer Spaces

Stefanos Angelidis[1] **Reinald Kim Amplayo**[1]
Yoshihiko Suhara[2] **Xiaolan Wang**[2] **Mirella Lapata**[1]
[1]University of Edinburgh [2]Megagon Labs
s.angelidis@ed.ac.uk, reinald.kim@ed.ac.uk
yoshi@megagon.ai, xiaolan@megagon.ai
mlap@inf.ed.ac.uk

Abstract

We present the Quantized Transformer (QT), an unsupervised system for extractive opinion summarization. QT is inspired by Vector-Quantized Variational Autoencoders, which we repurpose for popularity-driven summarization. It uses a clustering interpretation of the quantized space and a novel extraction algorithm to discover popular opinions among hundreds of reviews, a significant step towards opinion summarization of practical scope. In addition, QT enables controllable summarization without further training, by utilizing properties of the quantized space to extract aspect-specific summaries. We also make publicly available SPACE, a large-scale evaluation benchmark for opinion summarizers, comprising general and aspect-specific summaries for 50 hotels. Experiments demonstrate the promise of our approach, which is validated by human studies where judges showed clear preference for our method over competitive baselines.

1 Introduction

Online reviews play an integral role in modern life, as we look to previous customer experiences to inform everyday decisions. The need to digest review content has fueled progress in opinion mining (Pang and Lee, 2008), whose central goal is to automatically summarize people's attitudes towards an entity. Early work (Hu and Liu, 2004) focused on numerically aggregating customer satisfaction across different *aspects* of the entity under consideration (e.g., the quality of a camera, its size, clarity). More recently, the success of neural summarizers in the Wikipedia and news domains (Cheng and Lapata, 2016; See et al., 2017; Narayan et al., 2018; Liu et al., 2018; Perez-Beltrachini etal., 2019) has spurred interest in *opinion summarization*; the aggregation, in textual form, of opinions expressed in a set of reviews (Angelidis and Lapata, 2018; Huy Tien et al., 2019; Tian et al., 2019; Coavoux et al., 2019; Chu and Liu, 2019; Isonuma et al., 2019; Bražinskas et al., 2020; Amplayo and Lapata, 2020; Suhara et al., 2020; Wang et al., 2020).

Opinion summarization has distinct characteristics that set it apart from other summarization tasks. First, it cannot rely on reference summaries for training, because such *meta-reviews* are very scarce and their crowdsourcing is unfeasible. Even for a single entity, annotators would have to produce summaries after reading hundreds, sometimes thousands, of reviews. Second, the inherent subjectivity of review text distorts the notion of information *importance* used in generic summarization (Peyrard, 2019). Conflicting opinions are often expressed for the same entity and, therefore, useful summaries should be based on *opinion popularity* (Ganesan et al., 2010). Moreover, methods need to be *flexible* with respect to the size of the input (entities are frequently reviewed by thousands of users), and *controllable* with respect to the scope of the output. For instance, users may wish to read a *general* overview summary, or a more targeted one about a particular *aspect* of interest (e.g., a hotel's *location*, its *cleanliness*, or available *food* options).

Recent work (Tian et al., 2019; Coavoux et al., 2019; Chu and Liu, 2019; Isonuma et al., 2019; Bražinskas et al., 2020; Amplayo and Lapata, 2020; Suhara et al., 2020) has increasingly focused on *abstractive* summarization, where a summary is generated token-by-token to create novel sentences that articulate prevalent opinions in the input reviews. The abstractive approach offers a solution to the lack of supervision, under the assumption that opinion summaries should be written in the style of reviews. This simplification has allowed abstractive models to generate

Transactions of the Association for Computational Linguistics, vol. 9, pp. 277–293, 2021. https://doi.org/10.1162/tacl_a_00366
Action Editor: Jing Jiang. Submission batch: 6/2020; Revision batch: 9/2020; Published 3/2021.

review-like summaries from aggregate input representations, using sequence-to-sequence models trained to reconstruct single reviews. Despite being fluent, abstractive summaries may still suffer from issues of text degeneration (Holtzman et al., 2020), hallucinations (Rohrbach et al., 2018), and the undesirable use of first-person narrative, a direct consequence of review-like generation. In addition, previous work used an unrealistically small number of input reviews (10 or fewer), and only sparingly investigated controllable summarization, albeit in weakly supervised settings (Amplayo and Lapata, 2019; Suhara et al., 2020).

In this paper, we attempt to address shortcomings of existing methods by turning to *extractive* summarization, which aims to construct an opinion summary by selecting a few representative input sentences (Angelidis and Lapata, 2018; Huy Tien et al., 2019). Specifically, we introduce the *Quantized Transformer* (QT), an unsupervised neural model inspired by Vector-Quantized Variational Autoencoders (VQ-VAE; van den Oord et al., 2017; Roy et al., 2018), which we repurpose for popularity-driven summarization. QT combines Transformers (Vaswani et al., 2017) with the discretization bottleneck of VQ-VAEs and is trained via sentence reconstruction, similarly to the work of Roy and Grangier (2019) on paraphrasing. At inference time, we use a clustering interpretation of the quantized space and a novel extraction algorithm that discovers popular opinions among *hundreds of reviews*, a significant step towards opinion summarization of practical scope. QT is also capable of aspect-specific summarization without further training, by exploiting the properties of the Transformer's multi-head sentence representations.

We further contribute to the progress of opinion mining research, by introducing SPACE (shorthand for Summaries of Popular and Aspect-specific Customer Experiences), a large-scale corpus for the evaluation of opinion summarizers. We collected 1,050 human-written summaries of TripAdvisor reviews for 50 hotels. SPACE has *general* summaries, giving a high-level overview of popular opinions, and *aspect-specific* ones, providing detail on individual aspects (e.g., location, cleanliness). Each summary is based on 100 customer reviews, an order of magnitude increase over existing corpora, thus providing a more realistic input to competing models. Experiments on SPACE and two more benchmarks demonstrate that our approach holds promise for opinion summarization. Participants in human evaluation further express a clear preference for our model over competitive baselines. We make SPACE and our code publicly available.[1]

2 Related Work

Ganesan et al. (2010) were the first to make the connection between opinion mining and text summarization; they developed Opinosis, a graph-based abstractive summarizer that explicitly models *opinion popularity*, a key characteristic of subjective text, and central to our approach. Follow-on work (Di Fabbrizio et al., 2014) adopts a hybrid approach where salient sentences are first extracted and abstracts are generated based on hand-written templates (Carenini et al., 2006). More recently, Angelidis and Lapata (2018) extract *salient* opinions according to their polarity intensity and aspect specificity, in a weakly supervised setting.

A popular approach to modeling opinion popularity, albeit indirectly, is vector averaging. Chu and Liu (2019) propose MeanSum, an unsupervised abstractive summarizer that learns a review decoder through reconstruction, and uses it to generate summaries conditioned on averaged representations of the inputs. Averaging is also used by Bražinskas et al. (2020), who train a copy-enabled variational autoencoder by reconstructing reviews from averaged vectors of reviews about the same entity. Other methods include denoising autoencoders (Amplayo and Lapata, 2020) and the system of Coavoux et al. (2019), an encoder-decoder architecture that uses a clustering of the encoding space to identify opinion groups, similar to our work.

Our model builds on the VQ-VAE (van den Oord et al., 2017), a recently proposed training technique for learning discrete latent variables, which aims to overcome problems of posterior collapse and large variance associated with Variational Autoencoders (Kingma and Welling, 2014). Like other related discretization techniques (Maddison et al., 2017; Kaiser and Bengio, 2018), VQ-VAE passes the encoder output through a *discretization bottleneck* using a neighbor lookup in the space of latent code embeddings. The application of VQ-VAEs to opinion summarization is novel, to our knowledge, as well as the proposed sentence

[1] https://github.com/stangelid/qt.

extraction algorithm. Our model does not depend on vector averaging, nor does it suffer from information loss and hallucination. Furthermore, it can easily accommodate a large number of input reviews. Within NLP, VQ-VAEs have been previously applied to neural machine translation (Roy et al., 2018) and paraphrase generation (Roy and Grangier, 2019). Our work is closest to Roy and Grangier (2019) in its use of a quantized Transformer, however we adopt a different training algorithm (Soft EM; Roy et al., 2018), orders of magnitude fewer discrete latent codes, a different method for obtaining head sentence vectors, and apply the QT in a novel way for extractive opinion summarization.

Besides modeling, our work contributes to the growing body of resources for opinion summarization. We release SPACE, the first corpus to contain both general and aspect-specific opinion summaries, while increasing the number of input reviews tenfold compared to popular benchmarks (Bražinskas et al., 2020; Chu and Liu, 2019; Angelidis and Lapata, 2018).

3 Problem Formulation

Let C be a corpus of reviews on entities $\{e_1, e_2, \dots\}$ from a single domain d, for example, hotels. Reviews may discuss any number of relevant aspects $A_d = \{a_1, a_2, \dots\}$, like the hotel's *rooms* or *location*. For every entity e, we define its review set $R_e = \{r_1, r_2, \dots\}$. Every review is a sequence of sentences $(x_1, x_2, \dots)$ and a sentence x is, in turn, a sequence of words $(w_1, w_2, \dots)$. For brevity, we use X_e to denote all review sentences about entity e. We formalize two sub-tasks: (a) *general opinion summarization*, where a summary should cover popular opinions in R_e across all discussed aspects; and (b) *aspect opinion summarization*, where a summary must focus on a single specified aspect $a \in A_d$. In our extractive setting, these translate to creating a general or aspect summary by selecting a small subset of sentences $S_e \subset X_e$.

We train the QT through sentence reconstruction to learn a rich representation space and its quantization into latent codes (Section 3.1). We enable opinion summarization, by mapping input sentences onto their nearest latent codes and extract those sentences that are representative of the most popular codes (Section 3.2). We also illustrate how to produce aspect-specific summaries using a trained QT model and a few aspect-denoting query terms (Section 3.2.2).

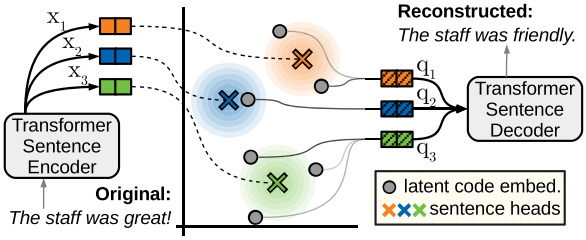

Figure 1: A sentence is encoded into a 3-head representation and head vectors are quantized using a weighted average of their neighboring code embeddings. The QT model is trained by reconstructing the original sentence.

3.1 The Quantized Transformer

Our model is a variant of VQ-VAEs (van den Oord et al., 2017; Roy and Grangier, 2019) and consists of: (a) a Transformer sentence *encoder* that encodes an input sentence x into a multi-head representation $\{\mathrm{x}_1, \dots, \mathrm{x}_H\}$, where $\mathrm{x}_h \in \mathbb{R}^D$ and H is the number of heads; (b) a vector *quantizer* that maps each head vector to a mixture of discrete latent codes, and uses the codes' embeddings to produce quantized vectors $\{\mathrm{q}_1, \dots, \mathrm{q}_H\}$, $\mathrm{q}_h \in \mathbb{R}^D$; and (c) a Transformer sentence *decoder*, which attends over the quantized vectors to generate sentence reconstruction $\hat{x}$. The decoder is not used during summarization; we only use the learned quantized space to extract sentences, as described in Section 3.2.

Sentence Encoding Our encoder prepends sentence x with the special token `[SNT]` and uses the vanilla Transformer encoder (Vaswani et al., 2017) to produce token-level vectors. We ignore individual word vectors and only keep the special token's vector $\mathrm{x}_{snt} \in \mathbb{R}^D$. We obtain a multi-head representation of x, by splitting x_{snt} into H sub-vectors $\{\mathrm{x}'_1, \dots, \mathrm{x}'_H\}$, $\mathrm{x}'_h \in \mathbb{R}^{D/H}$, followed by a layer-normalized transformation:

$$\mathrm{x}_h = \mathrm{LayerNorm}(\mathrm{W}\mathrm{x}'_h + \mathrm{b}), \qquad (1)$$

where x_h is the h-th head and $\mathrm{W} \in \mathbb{R}^{D \times D/H}$, $b \in \mathbb{R}^D$ are shared across heads. Hyperparameter H, that is, the number of *sentence heads* of our encoder, is different from Transformer's internal *attention heads*. The encoder's operation is illustrated in Figure 1, where the sentence "*The staff was great!*" is encoded into a 3-head representation.

Vector Quantization Let $z_1, \ldots, z_H$ be discrete latent variables corresponding to H encoder heads. Every variable can take one of K possible latent codes, $z_h \in [K]$. The quantizer's *codebook*, $\mathrm{e} \in \mathbb{R}^{K \times D}$, is shared across latent variables and maps each code (or *cluster*) to its embedding (or *centroid*) $\mathrm{e}_k \in \mathbb{R}^D$. Given sentence x and its multi-head encoding $\{\mathrm{x}_1, \ldots, \mathrm{x}_H\}$, we independently quantize every head using a mixture of its nearest codes from $[K]$. Specifically, we follow the *Soft EM* training of Roy et al. (2018) and sample, with replacement, m latent codes for the h-th head:

$$z_h^1, \ldots, z_h^m \sim \mathrm{Multinomial}(l_1, \ldots, l_K)\,, \quad \text{with } l_k = -\|\mathrm{x}_h - \mathrm{e}_k\|_2^2\,, \tag{2}$$

where $\mathrm{Multinomial}(l_1, \ldots, l_K)$ is a K-way multinomial distribution with logits $l_1, \ldots, l_K$. The h-th quantized head vector is obtained as the average of the sampled codes' embeddings:

$$\mathrm{q}_h = \frac{1}{m} \sum_{j=1}^{m} \mathrm{e}_{z_h^j}\,. \tag{3}$$

This soft quantization process is shown in Figure 1, where head vectors x_1, x_2, and x_3 are quantized using a weighted average of their neighboring code embeddings, to produce q_1, q_2, and q_3.

Sentence Reconstruction and Training Instead of attending over individual token vectors, as in the vanilla architecture, the Transformer sentence decoder attends over $\{\mathrm{q}_1, \ldots, \mathrm{q}_H\}$, the quantized head vectors of the sentence, to generate reconstruction $\hat{x}$. The model is trained to minimize:

$$L = L_r + \sum_h \|\mathrm{x}_h - \mathrm{sg}(\mathrm{q}_h)\|_2\,. \tag{4}$$

L_r is the reconstruction cross entropy, and stop-gradient operator $\mathrm{sg}(\cdot)$ is defined as identity during forward computation and zero on backpropagation. The sampling of Equation (2) is bypassed using the straight-through estimator (Bengio et al., 2013) and the latent codebook is trained via exponentially moving averages, as detailed in Roy et al. (2018).

3.2 Summarization in Quantized Spaces

Existing neural methods for opinion summarization have modeled opinion popularity within a set of reviews by encoding each review into a vector, averaging all vectors to obtain an aggregate representation of the input, and feeding it to a review decoder to produce a summary (Chu and Liu, 2019; Coavoux et al., 2019; Bražinskas et al., 2020). This approach is problematic for two reasons. First, it assumes that complex semantics of whole reviews can be encoded in a single vector. Second, it also assumes that features of commonly occurring opinions are preserved after averaging and, therefore, those opinions will appear in the generated summary. The latter assumption becomes particularly uncertain for larger numbers of input reviews.

We take a different approach, using sentences as the unit of representation, and propose a general extraction algorithm based on the QT, which explicitly models popularity without vector aggregation. Using the same algorithmic framework we are also able to extract aspect-specific summaries.

3.2.1 General Opinion Summarization

We exploit QT's quantization of the encoding space to cluster similar sentences together, quantify the popularity of the resulting clusters, and extract representative sentences from the most popular ones.

Specifically, given $X_e = \{x_1, \ldots, x_i, \ldots, x_N\}$, the N review sentences about entity e, the trained encoder produces $N \times H$ unquantized head vectors $\{\mathrm{x}_{11}, \ldots, \mathrm{x}_{ih}, \ldots, \mathrm{x}_{NH}\}$, where x_{ih} is the h-th head of the i-th sentence. We perform *hard* quantization, assigning every vector to its nearest latent code, and counting the number of assignments per code, namely, the *popularity* of each cluster:

$$z_{ih} = \arg\min_{k \in [K]} \|\mathrm{x}_{ih} - \mathrm{e}_k\|_2 \tag{5}$$

$$n_k = \sum_{i,h} \mathbb{1}[z_{ih} = k]\,. \tag{6}$$

Figure 2 shows how sentences X_e are encoded, and their different heads are assigned to codes. Similar sentences cluster under the same codes and, consequently, clusters receiving numerous assignments are characteristic of popular opinions in X_e. A general summary should consist of the sentences that are most *representative* of these popular clusters.

In the simplest case, we could couple every code k with its nearest sentence $x^{(k)}$:

$$x^{(k)} = \arg\min_i (\min_h \|\mathrm{x}_{ih} - \mathrm{e}_k\|_2)\,, \tag{7}$$

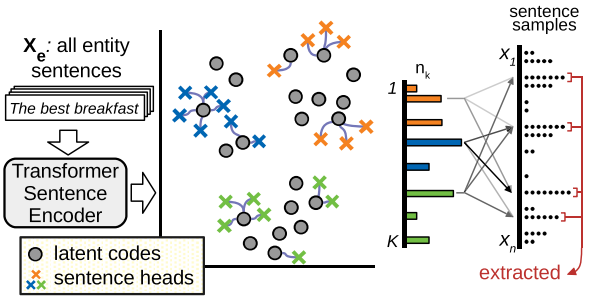

Figure 2: General opinion summarization with QT. All input sentences for an entity are encoded using three heads (shown in orange, blue, and green crosses). Sentence vectors are clustered under their nearest latent code (gray circles). Popular clusters (histogram) correspond to commonly occurring opinions, and are used to sample and extract the most representative sentences.

and rank sentences $x^{(k)}$ according to the size n_k of their respective clusters; the top sentences, up to a predefined budget, are extracted into a summary.

This ranking method entails that only those sentences which are the nearest neighbor of a popular code are likely to be extracted. However, a salient sentence may be in the neighborhood of multiple codes per head, despite never being the *nearest* sentence of a code vector. For example, the sentence *''Great location and beautiful rooms''* is representative of clusters encoding positive attitudes for both the *location* and the *rooms* of a hotel. To capture this, we relax the requirement of coupling every cluster with exactly one sentence and propose *two-step sampling* (Figure 3), a novel sampling process that simultaneously estimates cluster popularity and promotes sentences commonly found in the proximity of popular clusters. We repeatedly perform the following operations:

Cluster Sampling We first sample a latent code z with probability proportional to the clusters' size:

$$z \sim P(z = k) = \frac{n_k}{N \times H}\,, \tag{8}$$

where n_k is the number of assignments for code k, computed in Equation (6). For example, if the input contains many paraphrases of sentence *''Excellent location''*, these are likely to be clustered under the same code, which in turn increases the probability of sampling that code. Cluster sampling is illustrated on the top of Figure 3, showing

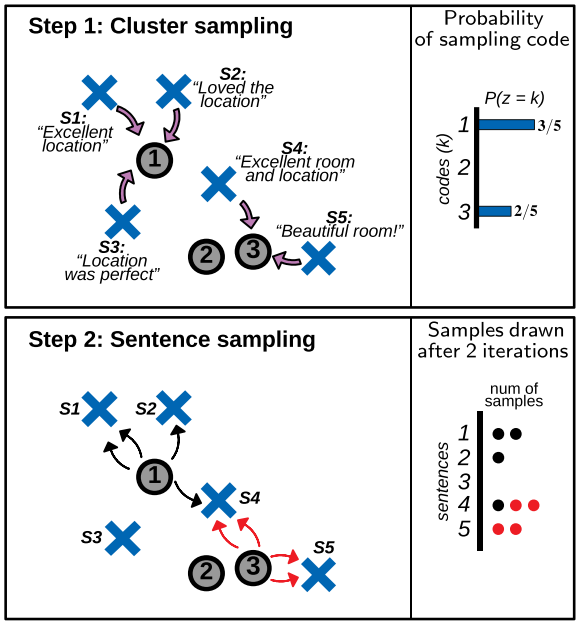

Figure 3: Sentence ranking via two-step sampling. In this toy example, each sentence (s_1 to s_5) is assigned to its nearest code ($k = 1, 2, 3$), as shown by thick purple arrows. During *cluster sampling*, the probability of sampling a code (top right; shown as blue bars) is proportional to the number of assignments it receives. For every sampled code, we perform *sentence sampling*; sentences are sampled, with replacement, according to their proximity to the code's encoding. Samples from codes 1 and 3 are shown in black and red, respectively.

assignments (left) and resulting code probabilities (right).

Sentence Sampling The sampled code z exists in the neighborhood of many input sentences. Picking a single sentence as the most characteristic of that cluster is too restrictive. Instead, we sample (with replacement) sentences from the code's neighborhood n times, thus generalizing Equation (7):

$$\begin{aligned} x^1, \ldots, x^n &\sim \text{Multinomial}(l'_1, \ldots, l'_N)\,, \\ \text{with } l'_i &= -\min_h(\|\mathrm{x}_{ih} - \mathrm{e}_z\|_2^2)\,, \end{aligned} \tag{9}$$

where the Multinomial's logits l'_i mark the (negative) distance of the i-th sentence's head which is closest to code z. Sentence sampling is depicted in the toy example of Figure 3 (bottom). After selecting code $k = 1$ during cluster sampling, four sentence samples are drawn (shown in black arrows). The next cluster sample ($k = 3$) results in four more sentence samples (shown in red).

Sentence s_4 (*"Excellent room and location"*) receives the most *votes* in total, after being sampled as a neighbor of both codes.

Two-step sampling is repeated multiple times and all sentences in X_e are ranked according to the total number of times they have been sampled. The final summary is constructed by concatenating the top ones (see right part of Figure 2). Importantly, our extraction algorithm is not sensitive to the size of the input. More sentences increase the absolute number of assignments per code, but do not hinder two-step sampling or cause information loss; on the contrary, a larger pool of sentences may result in a more densely populated quantized encoding space and, in turn, a better estimation of cluster popularity and sentence ranking.

3.2.2 Aspect Opinion Summarization

So far, we have focused on selecting sentences solely based on the popularity of the opinions they express. We now turn our attention to aspect summaries, which discuss a particular aspect of an entity (e.g., the *location* or *service* of a hotel) while still presenting popular opinions. We create such summaries with a trained QT model, without additional fine-tuning. Instead, we exploit QT's multi-head representations and only require a small number of aspect-denoting query terms.[2]

We hypothesize that different sentence heads in QT encode the approximately orthogonal semantic or structural attributes which are necessary for sentence reconstruction. In the simplified example in Figure 2, the encoder's first head (orange) might capture information about the aspects of the sentence, the second head (blue) encodes sentiment, while head three (green) may encode structural information (e.g., the length of the sentence or its punctuation). Our hypothesis is reinforced by the empirical observation that sentence vectors originating from the same head will occupy their own *sub-space*, and do not show any similarity to vectors from other heads. As a result, each latent code k receives assignments from exactly one head of the sentence encoder. More formally, head h yields a set of latent codes such that $K_h \subset [K]$. Figure 2 demonstrates this, as the encoding space consists of three sub-spaces, one for each head. Sentence and latent code vectors are

[2]Contrary to Angelidis and Lapata (2018), who used 30 seed-words per aspect, we only assume five query terms per aspect to replicate a realistic setting.

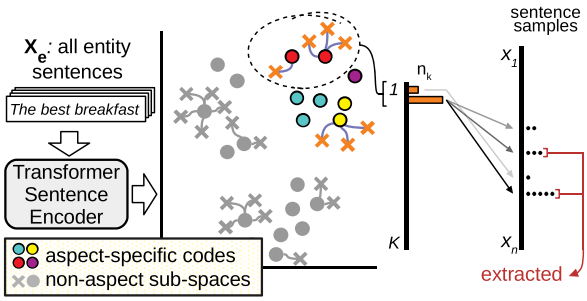

Figure 4: Aspect opinion summarization with QT. The aspect-encoding sub-space is identified using mean aspect entropy and all other sub-spaces are ignored (shown in gray). Two-step sampling is restricted only to the codes associated with the desired aspect (shown in red).

further organized within that sub-space according to the attribute captured by the respective head.

To enable aspect summarization, we identify the sub-space capturing aspect-relevant information and label its aspect-specific codes, as seen in Figure 4. Specifically, we first quantify the probability of finding an aspect in the sentences assigned to a latent code and identify the head sub-space that best separates sentences according to their aspect. Then, we map every cluster within that sub-space to an aspect and extract aspect summaries only from those aspect-specific clusters.

We utilize a held-out set of review sentences X_{dev}, and keywords $Q_a = \{s_1, \ldots, s_5\}$ for aspect a. We encode and quantize sentences in X_{dev} and compute the probability that latent code k contains tokens typical of aspect a as:

$$P_k(a) = \frac{tf(Q_a, k)}{\sum_{a'} tf(Q_{a'}, k)}, \qquad (10)$$

where $tf(Q_a, k)$ is the number of times query terms in Q_a where found in sentences assigned to k. We use information theory's *entropy* to measure how aspect-certain code k is:

$$\mathcal{H}_k = -\sum_a P_k(a) \log P_k(a). \qquad (11)$$

Low aspect entropy values indicate that most sentences assigned to k belong to a single aspect. It thus follows that h^{asp} (i.e., the head sub-space that best separates sentences according to their aspect) will exhibit the lowest mean aspect entropy:

$$h^{asp} = \arg\min_h \left(\frac{1}{|K_h|} \sum_{k \in K_h} \mathcal{H}_k \right). \qquad (12)$$

	Reviews	Entities	Rev/Ent	Summaries (R)	Type	Scope
SPACE (This work)	1.14M	50	100	1,050 (3)	Abstractive	General+Aspect
AMAZON (Bražinskas et al., 2020)	4.75M	60	8	180 (3)	Abstractive	General only
YELP (Chu and Liu, 2019)	1.29M	200	8	200 (1)	Abstractive	General only
OPOSUM (Angelidis and Lapata, 2018)	359K	60	10	180 (3)	Extractive	General only

Table 1: Statistics for SPACE and three recently introduced evaluation corpora for opinion summarization. SPACE includes aspect summaries for six aspects. (Reviews: number of reviews in *training* set, no gold-standard summaries are available; Rev/Ent: Input reviews per entity in *test set*; R: Reference summaries per example).

We map every code produced by h^{asp} to its aspect $a^{(k)}$ via Equation (10), and obtain aspect codes:

$$K_a = \{k \,|\, k \in K_{h^{asp}} \text{ and } a = a^{(k)}\}\,. \qquad (13)$$

To extract a summary for aspect a, we follow the ranking or sampling methods described in Equations (5)–(9), restricting the process to codes K_a. Sub-space selection and aspect-specific sentence sampling are illustrated in Figure 4.

4 The SPACE Corpus

We introduce SPACE (Summaries of Popular and Aspect-specific Customer Experiences), a large-scale opinion summarization benchmark for the evaluation of unsupervised summarizers. SPACE is built on TripAdvisor hotel reviews and aims to facilitate future research by improving upon the shortcomings of existing datasets. It comes with a training set of approximately 1.1 million reviews for over 11,000 hotels, obtained by cleaning and downsampling an existing collection (Wang et al., 2010). The training set contains no reference summaries, and is useful for unsupervised training.

For evaluation, we created a large collection of human-written, abstractive opinion summaries. Specifically, for a held-out set of 50 hotels (25 hotels for development and 25 for testing), we asked human annotators to write high-level general summaries *and* aspect summaries for six popular aspects: *building*, *cleanliness*, *food*, *location*, *rooms*, and *service*. For every hotel and summary type, we collected three reference summaries from different annotators. Importantly, for every hotel, summaries were based on 100 input reviews. To the best of our knowledge, this is the largest crowdsourcing effort towards obtaining high-quality abstractive summaries of reviews, and the first to use a pool of input reviews of this scale (see Table 1 for a comparison with existing datasets). Moreover, SPACE is the first benchmark to also contain aspect-specific opinion summaries.

The large number of input reviews per entity poses certain challenges with regard to the collection of human summaries. A direct approach is prohibitive, as it would require annotators to read all 100 reviews and write a summary in a single step. A more reasonable method is to first identify a subset of input sentences that most people consider salient, and then ask annotators to summarize them. Summaries were thus created in multiple stages using the Appen[3] platform and expert annotator channels of native English speakers. Although we propose an extractive model, annotators were asked to produce abstractive summaries, as we hope SPACE will be broadly useful to the summarization community. We did not allow the use of first-person narrative to collect more summary-like texts. We present our annotation procedure below.[4]

4.1 Sentence Selection via Voting

The sentence selection stage identifies a subset of review sentences that contain the most salient and useful opinions expressed by the reviewers. This is a crucial but subjective task and, therefore, we devised a voting scheme that allowed us to select sentences that received votes by many annotators.

Specifically, each review was shown to five judges who were asked to select informative sentences. Annotators were encouraged to exercise their own judgement in selecting summary-worthy sentences, but were advised to focus on sentences which explicitly expressed or supported reviewer opinions, avoiding overly general or personal comments (e.g., *''Loved the hotel''*, *''I like a shower with good pressure''*), and making sure that important aspects were included. We set no threshold on the number of sentences they could select (we allowed selecting *all* or *no* sentences

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

[4] Full annotation instructions: https://github.com/stangelid/qt/blob/main/annotation.md.

for a given review). However, the annotation interface kept track of their total votes and guided them to select between 20% and 40% of sentences, on average.

Sentences with 4 or more votes were automatically promoted to the next stage. Inter-annotator agreement according to Cohen's kappa was $k = 0.36$, indicating ''fair agreement''. Previous studies have shown that human agreement for sentence selection tasks in summarization of news articles is usually lower than 0.3 (Radev et al., 2003). The median number of sentences promoted for summarization for each hotel was 83, while the minimum was 46. This ensured that enough sentences were always available for summarization, while simplifying the task; annotators were now required to read and summarize considerably smaller amounts of review text than the original 100 reviews.

4.2 Summary Collection

General Summaries The top-voted sentences for each hotel were presented to three annotators, who were asked to read them and produce a high-level overview summary up to a budget of 100 words. To simplify the task and help annotators write coherent summaries, sentences with high lexical overlap were grouped together and the interface allowed the annotators to quickly sort sentences according to words they contained. The process resulted in an inter-annotator ROUGE-L score of 29.19 and provides ample room for future research, as detailed in our experiments (Table 2).

Aspect Summaries Top-voted sentences were further labeled by an off-the-shelf aspect classifier (Angelidis and Lapata, 2018) trained on an public aspect-labeled corpus of hotel review sentences (Marcheggiani et al., 2014).[5] Sentences outside of the six most popular aspects (*building*, *cleanliness*, *food*, *location*, *rooms*, and *service*) were ignored, and sentences with 3 votes were promoted, only if an aspect had no sentences with 4 votes. The promoted sentences were grouped according to their aspect and presented to annotators, who were asked to create a more detailed, aspect-specific summary, up to a budget of 75 words. The aspect summaries have an inter-annotator ROUGE-L score of 34.58.

[5]The classifier's precision on the aspect-labeled corpus' development set is 85.4%.

5 Evaluation

In this section, we discuss our experimental setup, including datasets and comparison models, before presenting our automatic evaluation results, human studies, and further analyses.

5.1 Experimental Setup

Datasets We used SPACE as the main testbed for our experimental evaluation, covering both general and aspect-specific summarization tasks. For general summarization, we used two additional opinion summarization benchmarks, namely, YELP (Chu and Liu, 2019) and AMAZON (Bražinskas et al., 2020) (see Table 1). For all datasets, we use pre-defined development and test set splits, and only report results on the test set.

Implementation Details We used unigram LM SentencePiece vocabularies of 32K.[6] All system hyperparameters were selected on the development set. The Transformer's dimensionality was set to 320 and its feed-forward layer to 512. We used 3 layers and 4 internal attention heads for its encoder and decoder, whose input embedding layer was shared, but no positional encodings as we observed no summarization improvements. We used $H = 8$ sentence heads for representing every sentence. For the quantizer, we set the number of latent codes to $K = 1,024$ and sampled $m = 30$ codes for every input sentence, during training. We used the Adam optimizer, with initial learning rate of 10^{-3} and a learning rate decay of 0.9. We warmed up the Transformer by disabling quantization for the first 4 epochs. In total, we ran 20 training epochs. On the full SPACE corpus, QT was trained in 4 days on a single GeForce GTX 1080 Ti GPU, using our available PyTorch implementation. All general and aspect summaries were extracted with the two-step sampling procedure described in Section 3.2.1, unless otherwise stated. When two-step sampling was enabled, we ranked sentences by sampling 300 latent codes and, for every code, sampled $n = 30$ neighboring sentences. QT and all extractive baselines use a greedy algorithm to eliminate redundancy, similar to previous research on multi-document summarization (Cao et al., 2015; Yasunaga et al., 2017; Angelidis and Lapata, 2018).

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

5.2 Metrics

We evaluate the lexical overlap between system and human summaries using ROUGE F-scores.[7] We report uni- and bi-gram variants (R1/R2), as well as longest common subsequence (RL).

A successful opinion summarizer must also produce summaries which match human-written ones in terms of aspects mentioned and sentiment conveyed. For this reason, we also evaluate our systems on two metrics that utilize an off-the-shelf aspect-based sentiment analysis (ABSA) system (Miao et al., 2020), pre-trained in-domain. The ABSA system extracts opinion phrases from summaries, and predicts their aspect category and sentiment. The metrics use these predictions as follows.

Aspect Coverage We use the phrase-level aspect predictions to mark the presence or absence of an aspect in a summary. We discard very infrequent aspect categories. Similar to Pan et al. (2020), we measure precision, recall, and F1 of system against human summaries.

Aspect-level Sentiment We propose a new metric to evaluate the sentiment consistency between system and human summaries. Specifically, we compute the sentiment polarity score towards an individual aspect a as the mean polarity of the opinion phrases that discuss this aspect in a summary ($pol_a \in [-1, 1]$). We repeat the process for every aspect, thus obtaining a vector of aspect polarities for the summary (we set the polarity of absent aspects to zero). The aspect-level sentiment consistency is computed as the mean squared error between system and human polarity vectors.

5.3 Results: General Summarization

We first discuss our results on general summarization and then move on to present experiments on aspect-specific summarization. We compared our model against the following baselines:

Best Review systems select the single review that best approximates the consensus opinions in the input. We use a *Centroid* method that encodes the entity's reviews with BERT (average token vector; Devlin et al., 2019) or SentiNeuron (Radford et al., 2017), and picks the one closest to the mean review vector. We also tested an *Oracle* method, which selects the review closest to the reference summaries.

Extractive systems, where we tested *LexRank* (Erkan and Radev, 2004), an unsupervised graph-based summarizer. To compute its adjacency matrices, we used BERT and SentiNeuron vectors, in addition to the sparse tf-idf features of the original. We also present a *random* extractive baseline.

Abstractive systems include *Opinosis* (Ganesan et al., 2010), a graph-based method; and *MeanSum* (Chu and Liu, 2019) and *Copycat* (Bražinskas et al., 2020), two neural abstractive methods that generate review-like summaries from aggregate review representations learned using autoencoders.

Table 2 reports ROUGE scores on SPACE (test set) for the general summarization task. QT's popularity-based extraction algorithm shows strong summarization capabilities outperforming all comparison systems (differences in ROUGE are statistically significant against all models but Copycat). This is a welcome result, considering that QT is an extractive method and does not benefit from the compression and rewording capabilities of abstractive summarizers. Moreover, as we discuss in Section 5.5, QT is less data-hungry than other neural models: It achieves the same level of performance even when trained on 5% of the dataset. We also show in Table 2 (fourth block) that the proposed two-step sampling method yields better extractive summaries compared to simply selecting the sentences nearest to the most popular clusters.

Aspect coverage and sentiment consistency results are also encouraging for QT which consistently scores highly on both metrics, while baselines show mixed results. We also compared (using ROUGE-L) *general* system summaries against reference *aspect* summaries. The results in Table 2 (column RL_{ASP}) confirm that aspect summarization requires tailor-made methods. Unsurprisingly, all systems are inferior to the human upper bound (i.e., inter-annotator ROUGE and aspect-based metrics), suggesting ample room for improvement.

QT's ability for general opinion summarization is further demonstrated in Table 3, which reports results on the YELP and AMAZON datasets. We present the strongest baselines, that is, $\text{Centroid}_{\text{BERT}}$, $\text{LexRank}_{\text{BERT}}$, $\text{Oracle}_{\text{BERT}}$, and the abstractive

[7] https://github.com/bheinzerling/pyrouge.

	SPACE [GENERAL]	R1	R2	RL	RL_{ASP}	AC_P	AC_R	AC_{F1}	SC_{MSE}
Best Review	$Centroid_{SENTI}$	27.36	5.81	15.15	8.77	.788	**.705**	.744	.580
	$Centroid_{BERT}$	31.33	5.78	16.54	9.35	.805	.701	.749	.524
	$Oracle_{SENTI}$	32.14	7.52	17.43	9.29	.817	.699	.753	.455
	$Oracle_{BERT}$	33.21	8.33	18.02	9.67	.823	.777	.799	.401
Extract	Random	26.24	3.58	14.72	11.53	.799	.374	.509	.592
	LexRank	29.85	5.87	17.56	11.84	.840	.382	.525	.518
	$LexRank_{SENTI}$	30.56	4.75	17.19	12.11	.820	.441	.574	.572
	$LexRank_{BERT}$	31.41	5.05	18.12	13.29	.823	.380	.520	.500
Abstract	Opinosis	28.76	4.57	15.96	11.68	.791	.446	.570	.561
	MeanSum	34.95	7.49	19.92	14.52	**.845**	.477	.610	.479
	Copycat	36.66	8.87	20.90	14.15	.840	.566	.676	.446
QT		**38.66**	**10.22**	**21.90**	14.26	.843	.689	**.758**	**.430**
	w/o 2-step samp.	37.82	9.13	20.10	13.88	.833	.680	.748	.439
Human Up. Bound		49.80	18.80	29.19	34.58	.829	.862	.845	.264

Table 2: Summarization results on SPACE. Best system (shown in **boldface**) significantly outperforms all comparison systems, except where underlined ($p < 0.05$; paired bootstrap resampling; Koehn, 2004). We exclude Oracle systems from comparisons as they access gold summaries at test time. RL_{ASP} is the Rouge-L of general summarizers against gold aspect summaries. AC and SC are shorthands for Aspect Coverage and Sentiment Consistency. Subscripts P and R refer to precision and recall, and F1 is their harmonic mean. MSE is mean squared error (lower is better).

YELP	R1	R2	RL	AC_{F1}	SC_{MSE}
Random	23.04	2.44	13.44	.551	.612
$Centroid_{BERT}$	24.78	2.64	14.67	.691	.523
$Oracle_{BERT}$	27.38	3.75	15.92	.703	.507
$LexRank_{BERT}$	26.46	3.00	14.36	.601	.541
Opinosis	24.88	2.78	14.09	.672	.552
MeanSum	28.46	3.66	15.57	.713	.510
Copycat	**29.47**	**5.26**	**18.09**	**.728**	.495
QT	28.40	3.97	15.27	.722	**.490**

AMAZON	R1	R2	RL	AC_{F1}	SC_{MSE}
Random	27.66	4.72	16.95	.580	.602
$Centroid_{BERT}$	29.94	5.19	17.70	.702	.599
$Oracle_{BERT}$	31.69	6.47	19.25	.725	.512
$LexRank_{BERT}$	31.47	5.07	16.81	.663	.541
Opinosis	28.42	4.57	15.50	.614	.580
MeanSum	29.20	4.70	18.15	.710	.525
CopyCat	31.97	5.81	**20.16**	.731	.510
QT	**34.04**	**7.03**	18.08	**.739**	**.508**

Table 3: Summarization results on YELP and AMAZON. Best system, shown in **boldface**, is significantly better than all comparison systems, except where underlined ($p < 0.05$; paired bootstrap resampling; Koehn, 2004).

Opinosis, MeanSum, and Copycat. On YELP, QT performs on par with MeanSum, but worse than Copycat. However, it is important to note that, in contrast to SPACE, YELP's reference summaries were purposely written using first-person narrative giving an advantage to review-like summaries of abstractive methods. On AMAZON, QT outperforms all methods on ROUGE-1/2, but comes second to Copycat on ROUGE-L. This follows a trend seen across all datasets, where abstractive systems appear relatively stronger in terms of ROUGE-L compared to ROUGE-1/2. We partly attribute this to their ability to fuse opinions into fluent sentences, thus matching longer reference sequences.

Besides automatic evaluation, we conducted a user study to verify the utility of the generated summaries. We produced general summaries from five systems (QT, Copycat, MeanSum, $LexRank_{BERT}$, and $Centroid_{BERT}$) for all entities in SPACE's test set. For every entity and pair of systems, we showed to three human judges a gold-standard summary for reference, and the two system summaries. We asked them to select the *best* summary according to four criteria:

	Inform.	Coherent	Concise	Redund.
Centroid	+36.0	−57.3	−60.7	−12.7
LexRank	−52.7	−38.0	−44.7	−1.3
MeanSum	−23.3	+26.7	+28.7	+3.3
Copycat	−10.7	+**34.7**	+38.0	−3.3
QT	+**50.7***	+34.0†	+**38.7**†	+**18.0***

Table 4: Best-Worst Scaling human study on SPACE. (*): significant difference to all models; (†): significant difference to all models, except Copycat (one-way ANOVA with posthoc Tukey HSD test $p < 0.05$).

informativeness (useful opinions, consistent with reference), *coherence* (easy to read, avoids contradictions), *conciseness* (useful in a few words), and *non-redundancy* (no repetitions). The systems' scores were computed using *Best-Worst Scaling* (Louviere et al., 2015), with values ranging from −100 (unanimously worst) to +100 (unanimously best). As shown in Table 4, participants rate QT favorably over all baselines in terms of informativeness, conciseness and lack of redundancy, with slight preference for Copycat summaries with respect to coherence (statistical significance information in caption). QT captures essential

	SPACE [ASPECT]	ROUGE-L *Building*	*Cleanliness*	*Food*	*Location*	*Rooms*	*Service*	R1	R2	RL	SC_{MSE}
								Average			
via BERT	$MeanSum_{ASP}$	13.25	19.24	13.01	18.41	17.81	20.40	23.24	3.72	17.02	.235
via BERT	$Copycat_{ASP}$	**17.10**	15.90	14.53	20.31	17.30	20.05	24.95	4.82	17.53	.274
via BERT	$LexRank_{ASP}$	14.73	25.10	17.56	23.28	18.24	26.01	27.72	7.54	20.82	.206
	QT_{ASP}	16.45	**25.12**	**17.79**	**23.63**	**21.61**	**26.07**	**28.95**	**8.34**	**21.77**	**.204**
	Human	40.33	38.76	33.63	35.23	29.25	30.31	44.86	18.45	34.58	.153

Table 5: Aspect summarization results on SPACE. Best model shown in **boldface**. All differences to best model are statistically significant, except where underlined ($p < 0.05$; paired bootstrap resampling; Koehn, 2004).

	Does the summary discuss the specified aspect? Exclusively	Partially	No
QT_{GEN}	1.1	72.0	26.9
$Copycat_{ASP}$	6.7	45.3	48.0
$MeanSum_{ASP}$	21.8	37.3	40.9
$LexRank_{ASP}$	48.2	28.0	23.8
QT_{ASP}	**58.7**	**32.7**	**8.7**

Table 6: User study on aspect-specific summaries. In the ''*Exclusively*'' column, QT's difference over all models is statistically significant ($p < 0.05$; χ^2 test).

SPACE [GENERAL]	**Proportion of SPACE's train data used** 5%	10%	50%	100%
Copycat	26.1	26.2	31.8	36.7
QT	36.9	37.1	37.7	38.7

Table 7: ROUGE-1 on SPACE for varying train set sizes.

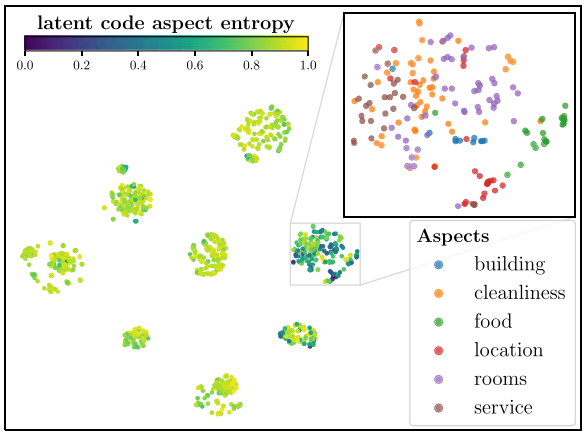

Figure 5: t-SNE projection of the quantized space of an eight-head QT trained on SPACE, showing all 1024 learned latent codes (best viewed in color). Darker codes correspond to lower *aspect entropy*, our proposed measure of high aspect-specificity. Zooming in the aspect sub-space uncovers good aspect separation.

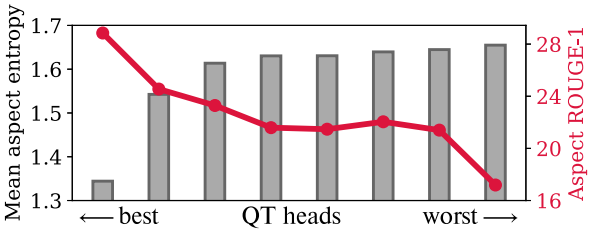

Figure 6: Mean aspect entropies (bars) for each of QT's head sub-spaces and corresponding aspect ROUGE-1 scores for the summaries produced by each head (line).

opinions effectively, whereas there is room for improvement in terms of summary cohesion.

5.4 Results: Aspect-specific Summarization

There is no existing unsupervised system for aspect-specific opinion summarization. Instead, we use the power of BERT (Devlin et al., 2019) to enable aspect summarization for our baselines. Specifically, we obtain BERT sentence vectors (average of token vectors) for input sentences X_e, which we cluster via k-means. We then replicate the cluster-to-aspects mapping used by QT, as described in Equations (10)–(13): Each cluster is mapped to exactly one aspect, according to the probability of finding the pre-defined aspect-denoting keywords in the sentences assigned to it. As a result, we obtain non-overlapping and aspect-specific sets of input sentences $\{X_e^{(a_1)}, X_e^{(a_2)}, \dots\}$. For aspect a_i, we create aspect-filtered input reviews, by concatenating sentences in $X_e^{(a_i)}$ based on the reviews they originated from. The filtered reviews of each aspect are given as input to general summarizers (LexRank, MeanSum, and Copycat), thus producing aspect summaries. QT and all baselines use the same aspect keywords, which we sourced from a held-out set of reviews, not included in SPACE.

Table 5 shows results on SPACE, for individual aspects, and on average. QT outperforms baselines

Human	QT	MeanSum	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.	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.	It was a great stay! The food at the hotel is great for the price. I can't believe the noise from the street is very loud and the traffic is not so great, but that is not a problem. The restaurant was great and the food is excellent.	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 8: Four general opinion summaries for the same hotel: One human-written and three from competing models.

Building: Bright colors, skateboards, butterfly chairs and a grand ocean/boardwalk view (always entertaining). There is a small balcony, but there's only a small glass divider between your neighbor's balcony.
Food: We had a great breakfast at Hash too! The restaurant was amazing. Lots of good restaurants within walking distance and some even deliver. The roof bar was the icing on the cake.
Location: The location is perfect. The hotel is very central. The hotel itself is in a great location. We hardly venture far as everything we need is within walking distance, but for the sightseers the buses are on the doorstep.
Cleanliness: Our room was very clean and comfortable. The room was clean and retrofitted with all the right amenities. Our room was very large, clean, and artfully decorated.
Rooms: The room was spacious and had really cool furnishings, and the beds were comfortable. The room's were good, and we had a free upgrade for one of them (for a Facebook 'like!) A+ for the bed and pillows.
Service: The staff is great. The staff were friendly and helpful. The hotel staff were friendly and provided us with great service. Each member of the staff was friendly and attentive. The staff excel and nothing is ever too much trouble.

Table 9: Aspect summaries extracted by QT.

in all aspects, except *building*, with significant improvements against Copycat and Meansum in terms of ROUGE and sentiment consistency. The abstractive methods struggle to generate summaries restricted to the aspect in question.

To verify this, we ran a second judgment elicitation study. We used summaries from competing aspect summarizers (QT_{ASP}, $Copycat_{ASP}$, $MeanSum_{ASP}$, and $LexRank_{ASP}$) for all six aspects, as well as QT's general summaries. A summary was shown to three participants, who were asked whether it discussed the specified aspect *exclusively*, *partially*, or *not at all*. Table 6 shows that 58.7% of QT aspect-specific summaries discuss the specified aspect exclusively, while only 8.7% of the summaries fail to mention the aspect. $LexRank_{ASP}$ follows with 23.8% of its summaries failing to mention the aspect, while the abstractive models performed significantly worse.

5.5 Further Analysis

Training Efficiency Table 7 shows ROUGE-1 scores for QT and Copycat on SPACE, when trained on different portions of the training set (randomly downsampled and averaged over 5 runs). QT exhibits impressive data efficiency; when trained on 5% of data, it performs comparably to a Copycat summarizer that has been trained on the full corpus.

Visualizing Sub-spaces We present a visual demonstration of QT's quantized sub-spaces in Figure 5. We used t-SNE (van der Maaten and Hinton, 2008) to project the latent code vectors onto two dimensions. The latent codes produced by QT's eight heads are clearly grouped in eight separate sub-spaces. The aspect sub-space (shown in square) was detected automatically, as it displayed the lowest mean aspect entropy (darker color). Zooming into its latent codes uncovers reasonable aspect separation, an impressive result considering that the model received no aspect-specific supervision.

Mean Aspect Entropy Figure 6 further illustrates the effectiveness of aspect entropy for detecting the head sub-space that best separates aspect-specific sentences. Each gray bar shows

the mean aspect entropy for the codes produced by one of QT's eight heads. One of the heads (leftmost) exhibits much lower entropy, indicating a strong confidence for aspect membership within its latent codes. We confirm this enables better aspect summarization by generating aspect summaries using each head, and plotting the obtained ROUGE-1 scores.

System Output Finally, we show gold-standard and system-generated general summaries in Table 8, as well as QT aspect summaries in Table 9.

6 Conclusions

We presented a novel opinion summarization system based on the Quantized Transformer that requires no reference summaries for training, and is able to extract general and aspect summaries from large groups of input reviews. QT is trained through sentence reconstruction and learns a rich encoding space, paired with a clustering component based on vector quantized variational autoencoders. At summarization time, we exploit the characteristics of the quantized space, to identify those clusters that correspond to the input's most popular opinions, and extract the sentences that best represent them. Moreover, we used the multi-head representations of the model, and no further training, to detect the encoding sub-space that best separates aspects, enabling aspect-specific summarization. We also collected SPACE, a new opinion summarization corpus which we hope will inform and inspire further research.

Experimental results on SPACE and popular benchmarks reveal that our system is able to produce informative summaries which cover all or individual aspects of an entity. In the future, we would like to utilize the QT framework in order to generate abstractive summaries. We could also exploit QT's multi-head semantics more directly, and further improve it through weak supervision or multi-task objectives. Finally, although we focused on opinion summarization, it would be interesting to see if the proposed model can be applied to other multi-document summarization tasks.

Acknowledgments

We thank the anonymous reviewers for their feedback and the action editor, Jing Jiang, for her comments. We gratefully acknowledge the support of the European Research Council (Lapata; award number 681760, ''Translating Multiple Modalities into Text''). We also thank Wang-Chiew Tan for her valuable input.

References

Reinald Kim Amplayo and Mirella Lapata. 2019. Informative and controllable opinion summarization. *arXiv preprint arXiv:1909.02322v1*.

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. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/2020.acl-main.175

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. **DOI:** https://doi.org/10.18653/v1/D18-1403

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv: 1308.3432v1*.

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. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/2020.acl-main.461

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2153–2159. AAAI Press.

Giuseppe Carenini, Raymond Ng, and Adam Pauls. 2006. Multi-document summarization of evaluative text. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

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. **DOI:** https://doi.org/10.18653/v1/P16-1046, **PMID:** PMC4738087

Eric Chu and Peter Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232, Long Beach, California, USA. 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. **DOI:** https://doi.org/10.18653/v1/D19-5405

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.

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. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/v1/W14-4408

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal Of Artificial Intelligence Research*, 22(1):457–479. **DOI:** https://doi.org/10.1613/jair.1523

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.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, Apr 26th - May 1st*. 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*, KDD '04, pages 168–177, New York, NY, USA. Association for Computing Machinery. **DOI:** https://doi.org/10.1145/1014052.1014073

Nguyen Huy Tien, Le Tung Thanh, and Nguyen Minh Le. 2019. Opinions summarization: Aspect similarity recognition relaxes the constraint of predefined aspects. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 487–496, Varna, Bulgaria. INCOMA Ltd. **DOI:** https://doi.org/10.26615/978-954-452-056-4_058, **PMID:** 30811828, **PMCID:** PMC6706286

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. **DOI:** https://doi.org/10.18653/v1/P19-1206

Lukasz Kaiser and Samy Bengio. 2018. Discrete autoencoders for sequence models. *ArXiv*, abs/1801.09797v1.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. *CoRR*, abs /1312.6114v10.

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. **DOI:** https://doi.org/10.1017/CBO9781107337855

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*. **DOI:** https://doi.org/10.1007/978-3-319-06028-6_23

Diego Marcheggiani, Oscar Täckström, Andrea Esuli, and Fabrizio Sebastiani. 2014. Hierarchical multi-label conditional random fields for aspect-oriented opinion mining. In *Advances in Information Retrieval*, pages 273–285, Cham. Springer International Publishing. **DOI:** https://doi.org/10.1007/978-3-319-06028-6_23

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*, WWW '20, pages 617–628, New York, NY, USA. Association for Computing Machinery. **DOI:** https://doi.org/10.1145/3366423.3380144

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. **DOI:** https://doi.org/10.18653/v1/N18-1158

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30*, pages 6306–6315. Curran Associates, Inc.

Haojie Pan, Rongqin Yang, Xin Zhou, Rui Wang, Deng Cai, and Xiaozhong Liu. 2020. Large scale abstractive multi-review summarization (LSARS) via aspect alignment. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 2337–2346, New York, NY, USA. Association for Computing Machinery. **DOI:** https://doi.org/10.1145/3397271.3401439

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2): 1–135. **DOI:** https://doi.org/10.1561/1500000011

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116, Florence, Italy. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P19-1504

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. **DOI:** https://doi.org/10.18653/v1/P19-1101

Dragomir R. Radev, Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Çelebi, Danyu Liu, and Elliott Drabek. 2003. Evaluation challenges in large-scale document summarization. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 375–382, Sapporo, Japan. Association for Computational Linguistics.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444v2*.

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. **DOI:** https://doi.org/10.18653/v1/D18-1437

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6033–6039, Florence, Italy. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P19-1605

Aurko Roy, Ashish Vaswani, Arvind Neelakantan, and Niki Parmar. 2018. Theory and experiments on vector quantized autoencoders. *arXiv preprint arXiv:1805.11063v2*.

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.

Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. 2020. OpinionDigest: A simple framework for opinion summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5789–5798, Online. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/2020.acl-main.513

Yufei Tian, Jianfei Yu, and Jing Jiang. 2019. Aspect and opinion aware abstractive review summarization with reinforced hard typed decoder. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pages 2061–2064, New York, NY, USA. Association for Computing Machinery. **DOI:** https://doi.org/10.1145/3357384.3358142, **PMID:** 30884086

L. J. P. van der Maaten and G.E. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(nov): 2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 783–792, New York, NY, USA. Association for Computing Machinery. **DOI:** https://doi.org/10.1145/1835804.1835903

Xiaolan Wang, Yoshihiko Suhara, Natalie Nuno, Yuliang Li, Jinfeng Li, Nofar Carmeli, Stefanos Angelidis, Eser Kandogann, and Wang-Chiew Tan. 2020. ExtremeReader: An interactive explorer for customizable and explainable review summarization. In *Companion Proceedings of the Web Conference 2020*, WWW '20, pages 176–180, New York, NY, USA.

Association for Computing Machinery. **DOI:** https://doi.org/10.1145/3366424.3383535

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/K17-1045