# Attributable and Scalable Opinion Summarization

**Tom Hosking** **Hao Tang** **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
tom.hosking@ed.ac.uk hao.tang@ed.ac.uk mlap@inf.ed.ac.uk

## Abstract

We propose a method for unsupervised opinion summarization that encodes sentences from customer reviews into a hierarchical discrete latent space, then identifies common opinions based on the frequency of their encodings. We are able to generate both abstractive summaries by decoding these frequent encodings, and extractive summaries by selecting the sentences assigned to the same frequent encodings. Our method is attributable, because the model identifies sentences used to generate the summary as part of the summarization process. It scales easily to many hundreds of input reviews, because aggregation is performed in the latent space rather than over long sequences of tokens. We also demonstrate that our appraoch enables a degree of control, generating aspect-specific summaries by restricting the model to parts of the encoding space that correspond to desired aspects (e.g., location or food). Automatic and human evaluation on two datasets from different domains demonstrates that our method generates summaries that are more informative than prior work and better grounded in the input reviews.

## 1 Introduction

Online review websites are a useful resource when choosing which hotel to visit or which product to buy, but it is impractical for a user to read hundreds of reviews. There has been significant interest in methods for automatically generating summaries or meta-reviews that aggregate the diverse opinions contained in a set of customer reviews about an *entity* (e.g., a product, hotel or restaurant) into a single summary.

Early work on opinion summarization extracted reviewers' sentiment about specific features (Hu and Liu, 2004) or selected salient sentences from reviews based on centrality (Erkan and Radev, 2004), while more recent methods based on neural models have used sentence selection in learned feature spaces (Angelidis et al., 2021; Basu Roy Chowdhury et al., 2022) or abstractive summarizers that generate novel output (Bražinskas et al., 2020, 2021; Amplayo et al., 2021a,b; Iso et al., 2021; Coavoux et al., 2019).

Following Ganesan et al. (2010), we define opinion summarization, or *review aggregation*, as the task of generating a textual summary that reflects frequent or popular opinions expressed in a large number of reviews about an entity. Systems are *extractive* if they select sentences or spans from the input reviews to use as the summary, or *abstractive* if they generate novel output. Review aggregation is challenging for a number of reasons. Firstly, it is difficult to acquire or create reference summaries, so models are almost always trained without access to gold standard references (Angelidis et al., 2021; Amplayo et al., 2021b, *inter alia*.). Secondly, popular entities may have hundreds of reviews, which can cause computational difficulties if the approach is not *scalable*. Finally, good summaries should be *abstractive* and not contain unnecessary detail, but should also not hallucinate false information. Ideally, a summarization system should be *attributable*, offering some evidence to justify its output.

Previous work has either been exclusively extractive (which is inherently attributable and often scalable but leads to unnecessarily specific summaries) or exclusively abstractive (which often scales poorly and hallucinates, e.g., Bražinskas et al., 2020) . We propose a hybrid method, that produces abstractive summaries accompanied by references to input sentences which act as evidence for each output sentence, allowing us to verify which parts of the input reviews were used to produce the output. Depicted in Figure 1, we first learn to encode natural language sentences from reviews as paths through a hierarchical discrete latent space. Then, given multiple review sentences about a specific entity, we identify common subpaths that

*Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*
*Volume 1: Long Papers*, pages 8488–8505
July 9-14, 2023 ©2023 Association for Computational Linguistics

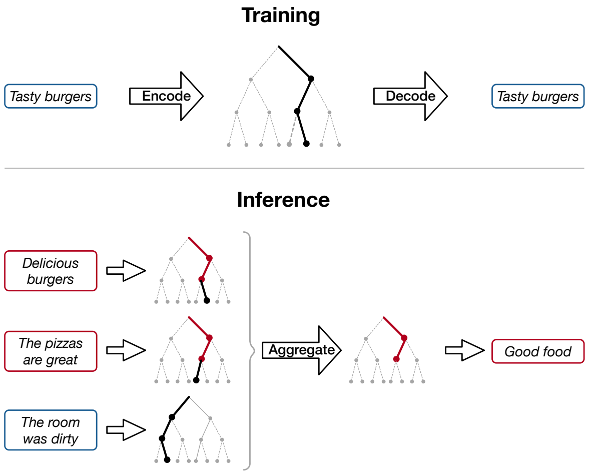

Figure 1: HERCULES is trained to encode sentences from reviews as paths through a hierarchical discrete latent space (top). At inference time, we encode all sentences from the input reviews, and identify frequent paths or subpaths to use for the summary (bottom). The consensus opinion from the three example inputs is that the food is good, so the subpath shown in red is repeated; decoding it should result in an output like "Good food".

are shared among many inputs, and decode them back to natural language, yielding the output summary. The sentences whose encodings contain the selected subpaths (shown in red in Figure 1) act as evidence for that generated sentence.

Our approach, HERCULES, is unsupervised and does not need reference summaries during training, instead relying on properties of the encoding space induced by the model. Since the aggregation process occurs in encoding space rather than over long sequences of tokens, HERCULES is highly scalable. Generated summaries are accompanied by supporting evidence from input reviews, making HERCULES attributable. It also offers a degree of controllability: we can generate summaries that focus on a specific aspect of an entity (e.g., location) or sentiment by restricting aggregation to subpaths that correlate with the desired property.

Our contributions are as follows:

- We propose a method for representing natural language sentences as paths through a hierarchical discrete latent space (Section 2).
- We exploit the properties of the learned hierarchy to identify common opinions from input reviews, and generate abstractive summaries alongside extractive *evidence sets* (Section 3).
- We conduct extensive experiments on two English datasets covering different domains, and show that our method outperforms previous state-of-the-art approaches, while offering the additional advantages of attributability and scalability (Sections 4 and 5).

## 2 Hierarchical Quantized Autoencoders

A good review aggregation system should identify frequent or common opinions, while abstracting away the details unique to a specific review. This joint requirement motivates our choice of a hierarchical discrete encoding: the discretization allows us to easily identify repeated opinions by counting them, while the hierarchy allows the model to encode high-level information (aspect, sentiment etc.) separately to specific details and phrasings.

### 2.1 Probabilistic Model

Let $\mathbf{y}$ be a sentence, represented as a sequence of tokens. We assume that the semantic content of $\mathbf{y}$ may be encoded as a set of discrete latent variables or *codes* $q_{1:D} \in [1, K]$. Further, we assume that the $q_{1:D}$ are ordered *hierarchically*, such that $q_1$ represents high level information about the sentence (e.g., the aspect or overall sentiment) whereas $q_D$ represents fine-grained information (e.g., the specific phrasing or choice of words used). The codes $q_{1:D}$ can be viewed as a single *path* through a hierarchy or tree as depicted in Figure 1, where each intermediate and leaf node in the tree corresponds to a sentence $\mathbf{y}$.

Thus, our generative model factorises as

$$p(\mathbf{y}) \quad = \quad \sum_{q_{1:D}} p(\mathbf{y}|q_{1:D}) \quad \times \quad \prod_{d=1}^{D} p(q_d) \qquad (1)$$

and the posterior factorises as

$$\phi(q_{1:D}|\mathbf{y}) \;=\; \phi(q_1|\mathbf{y}) \times \prod_{d=2}^{D} \phi(q_d|q_{<d}, \mathbf{y}). \qquad (2)$$

The training objective is given by

$$\begin{aligned} \text{ELBO} = \mathbb{E}_{\phi}\big[-\log p(\mathbf{y}|q_{1:D})\big] \\ + \beta_{KL} \sum_{d=1}^{D} D_{KL}\big[\phi(q_d|\mathbf{y}) \parallel p(q_d)\big] \qquad (3) \end{aligned}$$

where $q_d \sim \phi(q_d|\mathbf{y})$ and $\beta_{KL}$ determines the weight of the KL term. We choose a uniform prior for $p(q_d)$.

### 2.2 Neural Parameterization

The latent codes $q_{1:D}$ are discrete, but most neural methods operate in continuous space. We therefore need to define a mapping from the output $\mathbf{z} \in \mathbb{R}^{\mathbb{D}}$ of an encoder network $\phi(\mathbf{z}|\mathbf{y})$ to $q_{1:D}$, and vice versa for a decoder $p(\mathbf{y}|\mathbf{z})$. Similiar to Vector Quantization (VQ, van den Oord et al., 2017), we learn a codebook $\mathbf{C}_d \in \mathbb{R}^{K \times \mathbb{D}}$, which maps each discrete code to a continuous embedding $\mathbf{C}_d(q_d) \in \mathbb{R}^{\mathbb{D}}$.

Similar to HRQ-VAE (Hosking et al., 2022), since the $q_{1:D}$ are intended to represent hierarchical information, the distribution over codes at each level is a softmax distribution with scores $s_d$ given by the L2 distance from each of the codebook embeddings to the residual error between the input and the cumulative embedding from all previous levels,

$$s_d(q) = -\left(\left[\mathbf{x} - \sum_{d'=1}^{d-1} \mathbf{C}_{d'}(q_{d'})\right] - \mathbf{C}_d(q)\right)^2. \quad (4)$$

During inference, we set $q_d = \arg\max(s_d)$.

Given a path $q_{1:D}$, the input to the decoder $\mathbf{z}$ is given by the inverse of the decomposition process,

$$\mathbf{z} = \sum_{d=1}^{D} \mathbf{C}_d(q_d). \quad (5)$$

The embeddings at each level can be viewed as refinements of the (cumulative) embedding so far, or alternatively as selecting the centroid of a subcluster within the current cluster. Importantly, it is not necessary to specify a path to the complete depth $D$; a *subpath* $q_{1:d}$ $(d < D)$ still results in a valid embedding $\mathbf{z}$. We can therefore control the specificity of an encoding by varying its depth.

### 2.3 Training Setup

We use the Gumbel reparameterization (Jang et al., 2017; Maddison et al., 2017; Sønderby et al., 2017) to sample from the distribution over $q_{1:D}$. To encourage the model to explore the full codebook, we decay the Gumbel temperature $\tau$ according to the schedule given in Appendix A. We approximate the expectation in Equation (3) by sampling from the training set and updating via backpropagation (Kingma and Welling, 2014).

**Initialization Decay and Norm Loss** Smaller perturbations in encoding space should result in more fine-grained changes in the information they encode. Therefore, we encourage *ordering* between the levels of hierarchy (such that lower levels encode more fine-grained information) by initialising the codebook with a decaying magnitude, such that deeper embeddings have a smaller norm than those higher in the hierarchy. Specifically, the norm of the embeddings at level $d$ is weighted by a factor $(\alpha_{init})^{d-1}$. We also include an additional loss $\mathcal{L}_{NL}$ to encourage deeper embeddings to remain fine-grained during training,

$$\mathcal{L}_{NL} = \frac{\beta_{NL}}{D} \sum_{d=2}^{D} \left[ \max\left(\gamma_{NL} \frac{||\mathbf{C}_d||_2}{||\mathbf{C}_{d-1}||_2}, 1\right) - 1 \right]^2,$$

where $\gamma_{NL}$ determines the relative scale between levels and $\beta_{NL}$ controls the strength of the loss.

**Depth Dropout** To encourage the hierarchy within the encoding space to correspond to hierarchical properties of the output, we truncate at each level during training with some probability $p_{depth}$ (Hosking et al., 2022; Zeghidour et al., 2022). The output of the quantizer is then given by

$$\mathbf{z}_{syn} = \sum_{d=1}^{D} \left( \mathbf{C}_d(q_d) \prod_{d'=1}^{d} \gamma_{d'} \right), \quad (6)$$

where $\gamma_h \sim \text{Bernoulli}(1 - p_{depth})$. This means that the model is sometimes trained to reconstruct the output based only on a *partial* encoding of the input, and should learn to cluster similar outputs together at each level in the hierarchy.

**Denoising Objective** To encourage the model to group sentences according to their meaning rather than their syntactic structure, we use a denoising objective as a form of weak supervision. The model is trained to generate a target sentence from a different source sentence that has similar meaning but different surface form. For example, given the target sentence "We chose this hotel for price/location.", a source might be "I chose this hotel for its price and location.". The source sentences are retrieved automatically from other reviews in the training data using tf-idf (Jones, 1972) over bigrams; we select the top 5 most similar sentences for each target sentence with a minimum similarity of 0.6, and restrict to retrieving from reviews that have ratings equal to the target.

## 3 Aggregating Reviews in Encoding Space

So far, we have described a method for mapping from a sentence $\mathbf{y}$ to a path $q_{1:D}$ and vice versa.

We can now exploit the hierarchical property of the latent space to generate summaries.

Recall that the goal of review aggregation is to identify the majority or frequent opinions from a set of diverse inputs. This corresponds to identifying paths (or subpaths) in encoding space that are shared among many inputs. A simplified version of this process is depicted in the lower block of Figure 1; each sentence $\mathbf{y}^{(i)}$ in the input reviews is mapped to a path $q_{1:D}^{(i)}$ through the latent space. Summarizing these sentences is then reduced to the task of selecting a set of common subpaths, e.g., the subpath highlighted in red in Figure 1, which is shared between two out of three inputs.

**Subpath Selection** A simple approach would be to select the most frequent subpaths, but this would almost always result in high-level paths with $d = 1$ being selected (since every occurrence of a path $q_{1:d}$ entails an occurrence of all subpaths $q_{1:d'}, d' < d$). In practice there is a tradeoff between frequency and specificity. Additionally, good summaries often exhibit structure; they generally include high-level comments, alongside more specific comments about details that particularly differentiate the current entity from others. Indeed, some datasets (e.g., AmaSum, Bražinskas et al., 2021, Section 4.1) were constructed by scraping overall 'verdicts' and specific 'pros and cons' from review websites. We therefore reflect this structure and propose both a 'generic' and 'specific' method for selecting subpaths.

To select *generic* subpaths, we construct a probability tree from the set of input sentence encodings, with the node weights set to the observed path frequency $p(q_{1:d})$. Then, we iteratively prune the tree, removing the lowest probability leaves until all leaf weights exceed a threshold, $\min\big(p(q_{1:d})\big) > 0.01$. Finally, we select the leaves with the top $k$ weights to use for the summary. Empirically, this approach often selects paths with depth $d = 1$, but allows additional flexibility when a deeper subpath is particularly strongly represented.

Similar to Iso et al. (2022) we argue that the *specific* parts of the summary should also be comparative, highlighting details that are unique to the current entity. Thus, tf-idf (Jones, 1972) is a natural choice; we treat each path (and all its parent subpaths) as terms. We assign scores to each subpath $q_{1:d}$ proportional to its frequency within the current entity, and inversely proportional to the number of entities in which the subpath appears,

$$\text{score}(q_{1:d}) = \text{tf}(q_{1:d}) \times \log\big(\text{idf}(q_{1:d})\big). \quad (7)$$

Again, we select the subpaths with the top $k$ scores to use for the summary.

The overall summary is the combination of the selected generic and specific subpaths. The abstractive natural language output is generated by passing the selected subpaths as inputs to the decoder.

**Attribution** Each sentence in the generated summary has an associated subpath. By identifying all inputs which share that subpath, we can construct an *evidence set* of sentences that act as an explanation or justification for the generated output.

**Scalability** Since the aggregation is performed in encoding space, our method scales linearly with the number of input sentences (compared to quadratic scaling for Transformer methods that take a long sequence of all review sentences as input, e.g., Ouyang et al. (2022)), and can therefore handle large numbers of input reviews. In fact, since we identify important opinions using a frequency-based method, our system does not perform well when the number of input reviews is small, since there is no strong signal as to which opinions are common.

**Controlling the Output** Given an aspect $a$ (e.g., 'service') we source a set of keywords $\mathbb{K}_a$ (e.g., 'staff, friendly, unhelpful, concierge') associated with that aspect (Angelidis et al., 2021). We label each sentence in the training data with aspect $a$ if it contains any of the associated keywords $\mathbb{K}_a$, then calculate the probability distribution over aspects for each encoding path, $p(a|q_{1:D})$. We can modify the scoring function in Equation (7), multiplying the subpath scores during aggregation by the corresponding likelihood of a desired aspect, thereby upweighting paths relevant to that aspect,

$$\begin{aligned}\text{score}_{asp}(q_{1:d}) = \text{tf}(q_{1:d}) \times \log\big(\text{idf}(q_{1:d})\big) \\ \times\, p(a|q_{1:D}). \quad (8)\end{aligned}$$

We can also control for the sentiment of the summary; for the case where reviews are accompanied by ratings, we can label each review sentence (and its subpath) with the rating $r$ of the overall review, and reweight the subpath scores during aggregation by the likelihood of the desired rating $p(r|q_{1:D})$.

## 4 Experimental Setup

### 4.1 Datasets

We perform experiments on two datasets from two different domains. **SPACE** (Angelidis et al., 2021) consists of hotel reviews from TripAdvisor, with 100 reviews per entity. It includes reference summaries constructed by human annotators, with multiple references for each entity. It also includes reference *aspect-specific* summaries, which we use to evaluate the controllability of HERCULES.

**AmaSum** (Bražinskas et al., 2021) consists of reviews of Amazon products from a wide range of categories, with an average of 326 reviews per entity. The reference summaries were collected from professional review websites, and therefore are *not grounded* in the input reviews. The references in the original dataset are split into 'verdict', 'pros' and 'cons'; we construct single summaries by concatenating these three. We filter the original dataset down to four common categories (Electronics, Shoes, Sports & Outdoors, Home & Kitchen), and evaluate on a subset of 50 entities, training separate models for each. All systems were trained and evaluated on the same subsets.

### 4.2 Comparison Systems

We compare with a range of baseline and comparison systems, both abstractive and extractive. For comparison, we construct extractive summaries using HERCULES by selecting the centroid from each evidence set based on ROUGE-2 F1 score.

We select a **random review** from the inputs as a lower bound. We also select the **centroid** of the set of reviews, according to ROUGE-2 F1 score. We include an extractive **oracle** as an upper bound, by selecting the input sentence with highest ROUGE-2 similarity to each reference sentence.

**Lexrank** (Erkan and Radev, 2004) is an unsupervised extractive method using graph-based centrality scoring of sentences.

**QT** (Angelidis et al., 2021) uses vector quantization to map sentences to a discrete encoding space, then generates extractive summaries by selecting representative sentences from clusters.

**SemAE** (Basu Roy Chowdhury et al., 2022) is an extractive method that extends QT, relaxing the discretization and encoding sentences as mixtures of learned embeddings.

**CopyCat** (Bražinskas et al., 2020) is an abstractive approach that models sentences as observations of latent variables representing entity opinions.

**InstructGPT** (Ouyang et al., 2022) is a Large Language Model that generates abstractive summaries via prompting. We use the variant 'text-davinci-002'; training details are not public, but it is likely that it was tuned on summarization tasks, and potentially had access to the evaluation data for both SPACE and AmaSum during training.

**BiMeanVAE** and **COOP** (Iso et al., 2021) are abstractive methods that encode full reviews as continuous latent vectors, and take the average (BiMeanVAE) or an optimised combination (COOP) of review encodings.

Finally, for aspect specific summarization we compare to **AceSum** (Amplayo et al., 2021a). AceSum uses multi-instance learning to induce a synthetic dataset of review/summary pairs with associated aspect labels, which is then used to train an abstractive summarization model.

Most of the abstractive methods are not scalable and have upper limits on the number of input reviews. CopyCat and InstructGPT have a maximum input sequence length, while COOP exhaustively searches over combinations of input reviews. We use 8 randomly selected reviews as input to CopyCat and COOP, and 16 for InstructGPT.

### 4.3 Automatic Metrics

We use ROUGE F1 (Lin, 2004, R-2/R-L in Tables 1 and 2) to compare generated summaries to the references, calculated using the 'jackknifing' method for multiple references as implemented for the GEM benchmark (Gehrmann et al., 2021). To evaluate the faithfulness of the summaries, we use an automatic Question Answering (QA) pipeline inspired by Fabbri et al. (2022) and Deutsch et al. (2021): we use FlairNLP (Akbik et al., 2019) to extract adjectival- and noun-phrases from the *reference* summaries to use as candidate answers; we generate corresponding questions with a BART question generation model fine tuned on SQuAD (Lewis et al., 2020; Rajpurkar et al., 2016); finally, we attempt to answer these generated questions from the *predicted* summaries, using a QA model based on ELECTRA (Clark et al., 2020; Bartolo et al., 2021). We report the token F1 score of the QA model on the generated questions as 'QA'.

We also evaluate the extent to which the generated summaries are entailed by both the reference summaries and the input reviews using SummaC (Laban et al., 2022), reported as $SC_{\text{refs}}$ and $SC_{\text{in}}$ respectively. SummaC segments input reviews into

| | | SPACE | | | | | AmaSum (4 domains) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **System** | R-2 ↑ | R-L ↑ | QA ↑ | SC$_{\text{refs}}$ ↑ | SC$_{\text{in}}$ ↑ | R-2 ↑ | R-L ↑ | QA ↑ | SC$_{\text{refs}}$ ↑ | SC$_{\text{in}}$ ↑ |
| *Extractive* | Random | 6.16* | 17.13* | 9.92* | 25.97* | 50.02* | 1.02* | 9.46* | 3.09* | 22.41* | 59.17* |
| | Centroid | 7.68* | 17.79* | 14.17* | 25.82* | 53.99* | 2.00* | 11.21* | 3.89* | 23.47* | 64.63* |
| | LexRank | 5.87* | 16.42* | 8.64* | 22.63* | 51.31* | 2.66* | 12.20 | 4.95 | 23.49* | 67.20* |
| | QT | 10.28* | 21.50* | **17.12** | 41.15 | **90.78***| 1.51* | 11.41* | 3.62* | 22.42* | 66.21* |
| | SemAE | 11.12 | 23.48 | 10.03* | 27.89* | 59.67* | 1.58* | 11.26* | 2.66* | 21.83* | 57.19* |
| | HERCULES$_{\text{ext}}$ | **13.15** | **24.43** | 16.11 | **43.98** | 84.27 | **3.04** | **12.51** | **6.94** | **24.38** | **84.05** |
| *Abstractive* | CopyCat | 12.07* | 22.89* | 24.55 | 37.29* | 68.74* | 1.50* | 11.21* | 4.39* | 22.99* | 63.01* |
| | InstructGPT | 9.05* | 22.35* | 16.06* | 25.97* | 49.74* | 2.71* | 13.64* | 6.89 | 21.87* | 45.63* |
| | BiMeanVAE | 12.97* | 26.42 | 23.77 | 36.20* | 66.89* | 2.04 | 12.49 | 5.66 | 21.78* | 52.54* |
| | COOP | 13.53 | 26.56 | **25.21** | 39.35* | 70.26* | **2.79*** | **14.12*** | 6.14 | 22.51* | 58.35* |
| | HERCULES$_{\text{abs}}$ | **14.76** | **27.22** | 24.58 | **60.11** | **92.04** | 2.05 | 11.77 | **7.67** | **25.23** | **82.72** |
| | (References) | - | - | 93.62 | 93.43 | 58.90 | - | - | 89.40 | 86.68 | 65.59 |
| | (Oracle) | 45.02 | 53.29 | 31.74 | 69.59 | 64.14 | 14.36 | 26.04 | 14.04 | 26.38 | 76.35 |

Table 1: Results for automatic evaluation of summary generation. R-2 and R-L represent ROUGE-2/L F1 scores. QA indicates the F1 score of a question answering system attempting to answer questions generated from reference summaries, based on generated summaries. SC$_{\text{refs}}$ and SC$_{\text{in}}$ indicate degree of entailment (measured using SummaC) of generated summaries against reference summaries and input reviews respectively. Significant differences compared to each variant of HERCULES according to a paired t-test ($p < 0.05$) are marked with an asterisk, and best results in each class are bolded. Overall, both variants of HERCULES outperform comparison systems. In particular, summaries generated by HERCULES score highest on SC$_{\text{in}}$, indicating that they most strongly represent the information contained in the input reviews.

| | SPACE$_{\text{asp}}$ | | | | |
|---|---|---|---|---|---|
| **System** | R-2 ↑ | R-L ↑ | QA ↑ | SC$_{\text{refs}}$ ↑ | SC$_{\text{in}}$ ↑ |
| QT$_{\text{asp}}$ | 10.24 | 22.64 | 16.28 | 33.05 | **77.32** |
| AceSum$_{\text{ext}}$ | 12.10 | 27.15 | **20.15** | **38.04** | 67.48 |
| HERCULES$_{\text{ext}}$ | 7.93 | 19.96 | 13.84 | 26.12 | 66.64 |
| AceSum | **12.65** | **29.08** | 17.94 | 35.95 | 70.76 |
| HERCULES$_{\text{abs}}$ | 10.04 | 25.35 | 13.88 | 32.63 | 70.52 |
| (References) | - | - | 94.35 | 92.86 | 64.64 |

Table 2: ROUGE scores for controllable summarization, compared to the aspect-specific summaries in SPACE. Although not specifically designed for aspect-specific summarization, HERCULES is nonetheless able to generate useful summaries about a specified aspect.

sentence units and aggregates NLI scores between pairs of sentences to measure the strength of entailment between the source reviews and generated summary. SC$_{\text{in}}$ is the only *reference free* metric we use, and directly measures how well the generated summaries are supported by the input reviews. Since the references for AmaSum were constructed independently from the input reviews, we consider SC$_{\text{in}}$ to be our primary metric for AmaSum.

### 4.4 Model Configuration

We use a Transformer architecture (Vaswani et al., 2017) for our encoder $\phi(\mathbf{z}|\mathbf{x})$ and decoder $p(\mathbf{y}|\mathbf{z})$. Token embeddings were initialized from BERT (Devlin et al., 2019)[1]. We set the codebook size $K = 12$, with the number of levels $D = 12$, based on development set performance. Other hyperparameters are given in Appendix A. Our code and dataset splits are available at `https://github.com/tomhosking/hercules`.

For SPACE, we generate summaries using 5 generic and 5 specific paths (Section 3). For AmaSum, which was constructed from a single verdict sentence followed by more specific pros and cons, we use 1 generic path and 13 specific paths.

## 5 Results

**Automatic Evaluation** The results in Table 1 show that HERCULES outperforms previous approaches on both datasets. On SPACE, HERCULES$_{\text{abs}}$ achieves the highest ROUGE scores by some distance, and performs very well on all faithfulness metrics.

On AmaSum, HERCULES$_{\text{ext}}$ achieves higher ROUGE scores than HERCULES$_{\text{abs}}$; since the abstractive summaries are generated solely from the encodings, the decoder can sometimes mix up product types with similar descriptions (e.g., headphones and speakers) and is penalized accordingly. Since the references were not created from the input reviews, ROUGE scores are very low for all systems, and SC$_{\text{in}}$ is the most informative metric; both variants of HERCULES achieve the highest scores. Surprisingly, a number of the systems achieve SC$_{\text{in}}$ scores higher than the references, indicating that they are generating summaries that are

[1] We experimented with using BERT as the encoder but found no significant improvement, since the discrete encoding is the main bottleneck in the model.

| System | Output |
|---|---|
| *Reference* | The staff were very friendly, spoke fluent English, and helped with our local transportation needs and restaurant recommendations. The entire hotel was very clean, and the rooms and bathrooms were cleaned every day. The room was of good size for Paris and included a balcony. The bathroom was good sized, fully equipped, and private. Breakfast was continental and perfectly adequate. The location is good. |
| HERCULES$_{\text{ext}}$ | The room was very small. The staff is very friendly and helpful. It is walking distance to the highlights of the Latin quarter but a few blocks away from the college crowd (a good thing). The rooms were clean. The breakfast was sparse in choices. The location was great, being close to the place Monge Metro station. Breakfast was served in the basement. The bathroom was clean. They spoke English. The cafe across the street was yummy. |
| HERCULES$_{\text{abs}}$ | The room was clean and comfortable. The staff was very friendly and helpful. Walking distance to everything. Breakfast was good. The hotel is in a great location, just a few minutes walk from the train station. Breakfast was fine. The room and bathroom were very clean. The staff spoke English and were very helpful. There is also a small restaurant on the ground floor. |

Table 3: HERCULES output summaries convey useful information without being overly specific or verbose.

| Aspect | Output |
|---|---|
| *Rooms* | The room was very small. We had a room facing the street. The room was dark and dingy. The room and bathroom were very clean. |
| *Food* | The coffee was undrinkable. The breakfast was a bit disappointing. There is also a small restaurant on the ground floor. Breakfast is served in the basement. |
| *Location* | The hotel is in a great location, just a few minutes walk from the train station. The hotel is very basic. There is also a small restaurant on the ground floor. The location is very convenient. |

Table 4: Aspect-specific summaries from HERCULES$_{\text{abs}}$ convey information specific to the desired topic.

| | System | Info ↑ | Cohe ↑ | Conc ↑ |
|---|---|---|---|---|
| *Extractive* | Random | -9.68 | 0.20 | -3.21 |
| | LexRank | -10.14 | -22.31 | -24.54 |
| | QT | -8.05 | -5.44 | 0.51 |
| | HERCULES$_{\text{ext}}$ | **-0.10** | **-1.99** | **2.53** |
| | (References) | 30.00 | 30.15 | 24.12 |
| *Abstractive* | Random | -20.67 | -4.44 | -6.56 |
| | InstructGPT | 0.22 | **3.78** | **9.44** |
| | COOP | -8.00 | -12.78 | -9.56 |
| | HERCULES$_{\text{abs}}$ | **1.44** | -12.22 | -12.00 |
| | (References) | 29.17 | 27.33 | 19.33 |

Table 5: Results of our human evaluation. Crowdworkers were asked for pairwise preferences between generated summaries in terms of their informativeness (Info), coherence & fluency (Cohe) and conciseness & non-redundancy (Conc). Higher scores are better, and best values within each system type are bolded (excluding references). Overall, HERCULES generates more informative summaries than comparison systems.

more grounded in the inputs than the gold standard. Systems that model the summary as a single sequence, like InstructGPT and COOP, achieve high ROUGE-L scores because they generate very fluent output, but are less informative and less grounded in the input reviews according to SC$_{\text{in}}$, with InstructGPT scoring lowest on both datasets. Table 3 shows an example of a summary generated by HERCULES for an entity from SPACE. It covers a wide range of aspects, conveying useful information without being overly specific or verbose. We report additional examples in Appendix D.

To evaluate the controllability of HERCULES, we report the results of aspect-specific summarization on SPACE in Table 2 averaged across ‘rooms’, ‘location’, ‘cleanliness’, ‘building’, ‘service’ and ‘food’, with some example output shown in Table 4. Despite not being specifically trained or designed to generate aspect-specific summaries, HERCULES$_{\text{abs}}$ achieves reasonable scores across the range of metrics, and achieves comparable SC$_{\text{in}}$ scores to AceSum. Table 4 shows examples of aspect-specific summaries generated by HERCULES$_{\text{abs}}$, for the same entity. No sentiment-controlled reference summaries are available, but we show examples of sentiment-controlled output in Table 13. We conclude that HERCULES allows us to control the output of the model and generate summaries which focus on a specific aspect.

**Human Evaluation** Our goal is to generate summaries of hundreds of user reviews, but this makes human evaluation very difficult; it is not feasible to ask humans to keep track of the opinions expressed in hundreds of reviews. We are therefore limited to evaluation based on the references, but this is highly dependent on the reference quality. For AmaSum in particular the references are not grounded in the input reviews, and so the human evaluation is only indicative.

We recruited crowdworkers through Amazon Mechanical Turk, showed them a reference sum-

| Ablation | SPACE R-2 ↑ | SPACE R-L ↑ | SPACE SC$_{in}$ ↑ | AmaSum R-2 ↑ | AmaSum R-L ↑ | AmaSum SC$_{in}$ ↑ |
|---|---|---|---|---|---|---|
| HERCULES$_{abs}$ | 14.76 | 27.22 | 92.04 | 2.05 | 11.77 | 82.72 |
| No norm loss | -1.32 | -1.02 | -1.86 | -0.01 | -0.11 | +1.08 |
| No denoising | -1.99 | -2.85 | -5.34 | -0.17 | -0.23 | -7.75 |
| Generic only | -0.82 | -0.59 | +3.28 | -0.66 | -0.70 | -14.77 |
| Specific only | -1.49 | -3.41 | -11.24 | -1.15 | -2.18 | -9.91 |
| VAE + k-means | -2.77 | -3.71 | -34.14 | -1.12 | -2.54 | -1.70 |

Table 6: Changes in key metrics for a range of ablations of the HERCULES$_{abs}$ model. Removing the components tested leads to a drop in performance.

| | |
|---|---|
| *Output* | Breakfast was good. |
| *Evidence* | Breakfast was very good for us<br>Breakfast offers a variety of things to eat.<br>The buffet breakfast is varied and satisfying<br>The buffet breakfast was all fresh food with a good choice<br>Breakfast was good. |
| *Output* | Great camera for the price. |
| *Evidence* | I like the camera.<br>Overall a great camera at a good price.<br>I like the range of the lens.<br>Great camera.<br>This is a good camera for the money. |

Table 7: Examples of evidence sets produced by HERCULES. Each output sentence generated by the model is attached to a set of input sentences that share the same subpath.

| | |
|---|---|
| *Input* | The staff was very helpful; the free breakfast was the best we had on this trip. |
| *Output* ($d = 1$) | Breakfast was good. |
| *Output* ($d = 2$) | The continental breakfast was a joke. |
| *Output* ($d = 3$) | The breakfast was one of the best I have ever had. |
| *Output* ($d = 4$) | The breakfast was one of the best I've had in a hotel. |
| *Cluster* ($d = 3$) | Continental breakfast was the BEST so far on our trip!!!<br>The staff was very helpful; the free breakfast was the best we had on this trip.<br>The Cafe has the among the best breakfast and lunch in Vegas (closed for dinner). |

Table 8: An example of how our model encodes sentences at different granularities. As more levels are used, the output increasingly converges towards the meaning expressed by the input. We also show other input sentences that are assigned the same subpath (of depth = 3); despite very different phrasing, they convey a common opinion.

mary alongside two generated summaries, and solicited pairwise preferences along three dimensions: Informativeness, Conciseness & Non-Redundancy, and Coherence & Fluency. The full instructions are reproduced in Appendix B. We gathered annotations for all 25 entities in the SPACE test set and 10 entities from each AmaSum domain, with 3 annotations for each. Extractive and abstractive systems were evaluated separately. The results in Table 5 show that both variants of HERCULES produce summaries that are considered to be more informative than other systems, although this comes at the cost of slightly lower coherence.

**Attribution** Since our approach is attributable and produces evidence sets alongside each abstractive summary sentence, we can evaluate the degree to which the generated sentences are supported by the evidence they cite. We used SummaC to measure the strength of entailment between each generated sentence and its evidence set, giving scores of 71.2 for SPACE and 46.8 for AmaSum. We also performed a human evaluation on a random sample of 150 output sentences, and found that generated sentences were supported by the majority of the associated evidence set 65% of the time for SPACE and 57.3% for AmaSum. We invite future work to facilitate this kind of evaluation and to improve on our level of factuality.

**Ablations** To evaluate to the contribution of each component towards the overall performance, we perform a range of ablation studies. Table 6 shows the changes in key metrics for models trained without the norm loss and without the denoising objective. We also evaluate summaries generated using only the generic and specific subpath selection methods, rather than a combination of both. Finally, we evaluate the importance of learning the clusters at the same time as the model, rather than post-hoc: we train a model with the same training data and hyperparameters as HERCULES but a *continuous* encoding; use k-means clustering over sentence encodings to identify a set of centroids for each entity; and finally generate a summary by passing the centroids to the decoder. The results show that all components lead to improved summary quality. The centroids extracted from a continuous VAE using k-means may not necessarily correspond to a valid sentence, leading to poor quality output.

**Analysis** Table 7 shows examples of evidence sets, illustrating how HERCULES is able to generate output that retains key information from the inputs, while discarding unnecessary detail. Table 8 shows a breakdown of generated output at different granularities. Given the input sentence, we show the output of the decoder with subpaths of varying granularities, demonstrating how subpaths of increasing depth lead to more detailed output.

Figure 2 shows a t-SNE (van der Maaten and

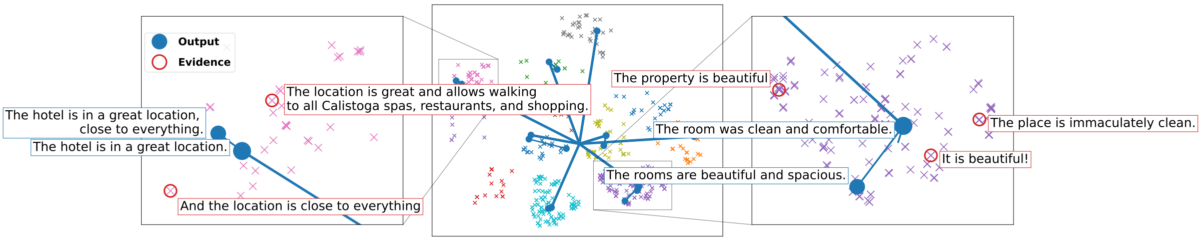

Figure 2: A t-SNE (van der Maaten and Hinton, 2008) plot of the embeddings of all review sentences from a single entity from SPACE, where the colour of the points represents the top level code $q_1$. The summary subpaths are overlaid in blue, alongside output from different hierarchy depths. A selection of evidential inputs are circled in red.

Hinton, 2008) plot of the embeddings of all review sentences for a single entity from SPACE, with the summary subpaths overlaid on top in blue. We include a more detailed view of two summary subpaths (left and right panels), showing the increasing level of detail as more levels are specified. We also highlight sample input sentences from the evidence set (circled in red), demonstrating how the generated output can be attributed to input sentences conveying similar opinions.

HERCULES is trained to reconstruct a target sentence from a source retrieved using tf-idf, but tf-idf is not sensitive to negation and does not distinguish between syntax and semantics. We observe that the model sometimes clusters sentences with superficially similar surface forms but different meanings. For example, “The breakfast buffet was very good” and “The breakfast buffet was not very good either” are assigned to the same path by our model.

The model is trained to generate output sentences based solely on the latent encoding: this is required to ensure that the model learns a useful encoding space. However, it also makes the model susceptible to some types of hallucination. Sentences about similar topics are likely to be assigned to the same paths, so the model may generate output that mentions a different entity of similar type (e.g., headphones instead of speakers).

## 6 Related work

Previous work has investigated aggregating user opinions in a latent space, but these approaches have generally been purely extractive for discrete spaces (Angelidis et al., 2021; Basu Roy Chowdhury et al., 2022) or purely abstractive for continuous spaces (Iso et al., 2021). Other approaches have either been supervised (Bražinskas et al., 2021) or have selected ‘central’ reviews to use as proxy summaries for training (Amplayo et al., 2021a,b), but they do not explicitly model the aggregation process. Iso et al. (2022) propose a method for that highlights both common and contrastive opinions.

Hierarchical VQ was introduced by Juang and Gray (1982) as ‘multistage VQ’, with a set of codebooks fitted post-hoc to a set of encoding vectors, and further developed as ‘Residual VQ’ by Chen et al. (2010) and Xu et al. (2022). More recently, Zeghidour et al. (2022) and Hosking et al. (2022) concurrently proposed methods for learning the codebook jointly with an encoder-decoder model. A form of hierarchical VQ has also been proposed in computer vision (Razavi et al., 2019), but in their context the hierarchy refers to a stacked architecture rather than to the latent space. A separate line of work has looked at learning hierarchical latent spaces using hyperbolic geometry (Mathieu et al., 2019; Surís et al., 2021), but the encodings are still continuous and not easily aggregated.

The recent surge in performance of language models has led to a desire to evaluate whether the information they output is verifiable. Rashkin et al. (2021) propose a framework for post-hoc annotation of system output to evaluate attributability; we argue that it is better to have systems that justify their output as part of the generation process.

## 7 Conclusion

We propose HERCULES, a method for aggregating user reviews into textual summaries by identifying frequent opinions in a discrete latent space. Compared to previous work, our approach generates summaries that are more informative, while also scaling to large numbers of input reviews and providing evidence to justify its output.

Future work could combine the improvements in attributability and scalability of our model with the fluency of systems that model summaries as a single sequence. Allowing the model to access the evidence sets during decoding could lead to improved output quality with less hallucination.

## Limitations

Since our approach identifies common opinions based on frequency of sentence encodings, we require a relatively large number of input sentences. We were not able to experiment with other popular datasets like Amazon (He and McAuley, 2016), Yelp (Chu and Liu, 2019) or Rotten Tomatoes (Wang and Ling, 2016) since these datasets only include a small number (usually 8) of input reviews.

The abstractive summaries are generated solely based on the latent encoding, and our model does not include a copy mechanism or attend to the original inputs when decoding. It therefore does not always generalize well to new domains. However, this limitation is mitigated by not requiring any labelled data during training: HERCULES can easily be retrained on a new domain.

Generating output based only on latent encodings means that the model is also susceptible to hallucinating, since the output is less directly linked to the inputs. However, unlike other methods, HERCULES provides evidence sets alongside the generated summaries, making it easier to check whether the output is faithful.

Finally, HERCULES generates summary sentences independently, leading to summaries that are less coherent than approaches that model the summary as a single sequence. We welcome future work on combinining the relative strengths of each approach. We do not anticipate any significant risks resulting from this work.

## Acknowledgements

We thank our anonymous reviewers for their feedback. This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh. Lapata acknowledges the support of the UK Engineering and Physical Sciences Research Council (grant EP/W002876/1).

## References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

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 *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12489–12497. AAAI Press.

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.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 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.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2021. Learning opinion summarizers by selecting informative reviews. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9424–9442, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yongjian Chen, Guan Tao, and Cheng Wang. 2010. Approximate nearest neighbor search by residual vector quantization. *Sensors (Basel, Switzerland)*, 10:11259–73.

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.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.

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.

G. Erkan and D. R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

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.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

Ruining He and Julian 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 '16, page 507–517, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Tom Hosking, Hao Tang, and Mirella Lapata. 2022. Hierarchical sketch induction for paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2489–2501, Dublin, Ireland. Association for Computational Linguistics.

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, page 168–177, New York, NY, USA. Association for Computing Machinery.

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

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

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with Gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Biing-Hwang Juang and A. Gray. 1982. Multiple stage vector quantization for speech coding. In *ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 7, pages 597–600.

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

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

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

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

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Emile Mathieu, Charline Le Lan, Chris J. Maddison, Ryota Tomioka, and Yee Whye Teh. 2019. Continuous hierarchical representations with Poincaré variational auto-encoders. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12544–12555.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2021. Measuring attribution in natural language generation models. *CoRR*, abs/2112.12870.

Ali Razavi, Aäron van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14837–14847.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Casper Kaae Sønderby, Ben Poole, and Andriy Mnih. 2017. Continuous relaxation training of discrete latent variable image models. In *Beysian DeepLearning workshop, NIPS*, volume 201.

Dídac Surís, Ruoshi Liu, and Carl Vondrick. 2021. Learning the predictability of the future. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12602–12612.

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.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

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.

Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 47–57, San Diego, California. Association for Computational Linguistics.

Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2017. Data noising as smoothing in neural network language models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zhi Xu, Lushuai Niu, Ruimin Meng, Longyang Zhao, and Jianqiu Ji. 2022. Residual vector product quantization for approximate nearest neighbor search. In *Advances in Knowledge Discovery and Data Mining*, pages 208–220, Cham. Springer International Publishing.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2022. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 30:495–507.

Adrian Łańcucki, Jan Chorowski, Guillaume Sanchez, Ricard Marxer, Nanxin Chen, Hans J.G.A. Dolfing, Sameer Khurana, Tanel Alumäe, and Antoine Laurent. 2020. Robust training of vector quantized bottleneck models. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

## A Replication details

Models were trained on a single A100 GPU, with training taking roughly 24 hours for SPACE and 6 hours for each AmaSum domain.

The prompt used for InstructGPT evaluation was as follows:

> Review:
>
> [Review 1 text]
>
> Review:
>
> [Review 2 text]
>
> [...]
>
> Summarize these reviews:

Table 9 show the hyperparameters used for our experiments. The Gumbel temperature was decayed from $\tau_0$ to $\tau_{min}$ according to

$$\tau \quad = \quad \max\left(\tau_0 \times \exp(-\frac{t}{\gamma_{temp}}), \tau_{min}\right), \quad (9)$$

in line with Jang et al. (2017).

The model is sensitive to the initialization of the codebook; the initial embeddings should be located in roughly the same region of space as the output of the encoder, but should have sufficient variation so as to be informative for the decoder. Following Łańcucki et al. (2020) we initialize the codebook on a unit hypersphere, to avoid the radial distance component dominating the angular component.

| Parameter | Value |
|---|---|
| Embedding dim. $D$ | 768 |
| Encoder layers | 5 |
| Decoder layers | 5 |
| Feedforward dim. | 2048 |
| Transformer heads | 8 |
| Depth $D$ | 12 |
| Codebook size $K$ | 12 |
| Optimizer | Adam (Kingma and Ba, 2015) |
| Learning rate | 5e-4 |
| Batch size | 200 |
| Token dropout | 0.2 (Xie et al., 2017) |
| Decoder | Beam search |
| Beam width | 4 |
| $\alpha_{init}$ | 0.5 |
| $\tau_0$ | 1.0 |
| $\tau_{min}$ | 0.5 |
| $\gamma_{temp}$ | 33333 |
| $\beta_{KL}$ | 0.0025 |
| $\beta_{NL}$ | 0.05 |
| $\gamma_{NL}$ | 1.5 |

Table 9: Hyperparameter values used for our experiments.

We used the default settings for SummaC (Laban et al., 2022) as given on the project GitHub, using the SummaCConv variant trained on VitaminC (Schuster et al., 2021) and mean aggregation.

## B Human Evaluation

The instructions given to crowdworkers were as follows:

> In this task you will be presented with a number of summaries produced by different automatic systems based on user reviews. Your task is to select the best system summary based on the criteria listed below.
>
> Please read the human summary first and try to get an overall idea of what opinions it expresses.
>
> Please read the criteria descriptions and system summaries carefully, and whenever is necessary re-read the human summary.

> Remember that you are being asked to rate the system, not the human summary.
>
> **Informativeness** Which system summary gives useful information that is consistent with the opinions in the human summary?
>
> **Conciseness & Non-Redundancy** Which system summary includes useful information in a concise manner and avoids repetitions?
>
> **Coherence & Fluency** Which system summary is easy to read and avoids contradictions?

Crowdworkers were recruited from the UK and US, and were supplied with a Participant Information Sheet before being asked for their consent to participate. Crowdworkers were compensated $0.30 per annotation which took approximately 1.5 minutes, corresponding to an hourly wage of $12.00/hour. This exceeds the US federal minimum wage ($7.25) at time of writing.

## C Breakdown of Results

We report the automatic evaluation scores broken down by AmaSum domains in Table 10 and Table 11, and the human evaluation results broken down by dataset in Appendix C.

## D Example Output

Table 13 shows an example of generated summaries with sentiment control. We report additional examples of output summaries in Table 14.

| | System | Electronics | | Home/Kitchen | | Shoes | | Sports/Outdoors | |
|---|---|---|---|---|---|---|---|---|---|
| | | R-2 ↑ | R-L ↑ | R-2 ↑ | R-L ↑ | R-2 ↑ | R-L ↑ | R-2 ↑ | R-L ↑ |
| *Extractive* | Random | 0.95 | 9.51 | 1.09 | 9.78 | 0.76 | 8.41 | 1.29 | 10.17 |
| | Centroid | 1.78 | 11.53 | 2.50 | 12.47 | 1.63 | 9.43 | 2.07 | 11.41 |
| | LexRank | 2.47 | 12.18 | 3.22 | 12.84 | 1.96 | 10.51 | 3.00 | 13.29 |
| | QT | 1.55 | 10.95 | 1.79 | 12.15 | 1.23 | 11.13 | 1.46 | 11.43 |
| | SemAE | 1.32 | 10.97 | 2.37 | 12.95 | 1.32 | 9.64 | 1.32 | 11.48 |
| | HERCULES$_{\text{ext}}$ | 3.29 | 12.48 | 3.19 | 12.89 | 2.67 | 11.75 | 3.00 | 12.93 |
| *Abstractive* | CopyCat | 1.46 | 11.92 | 2.11 | 11.86 | 0.98 | 9.00 | 1.46 | 12.07 |
| | InstructGPT | 2.83 | 13.89 | 2.99 | 14.39 | 2.23 | 12.52 | 2.80 | 13.76 |
| | BiMeanVAE | 2.32 | 12.42 | 2.32 | 12.93 | 1.46 | 11.80 | 2.05 | 12.81 |
| | COOP | 3.46 | 14.56 | 2.66 | 14.22 | 2.78 | 13.39 | 2.28 | 14.31 |
| | HERCULES$_{\text{abs}}$ | 2.46 | 12.57 | 2.22 | 11.53 | 1.80 | 11.77 | 1.72 | 11.21 |

Table 10: Results for ROUGE scores with respect to references on AmaSum, broken down by product category.

| | System | Electronics | | | Home/Kitchen | | | Shoes | | | Sports/Outdoors | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | QA ↑ | SC$_{\text{refs}}$ ↑ | SC$_{\text{in}}$ ↑ | QA ↑ | SC$_{\text{refs}}$ ↑ | SC$_{\text{in}}$ ↑ | QA ↑ | SC$_{\text{refs}}$ ↑ | SC$_{\text{in}}$ ↑ | QA ↑ | SC$_{\text{refs}}$ ↑ | SC$_{\text{in}}$ ↑ |
| *Extractive* | Random | 1.90 | 22.55 | 57.27 | 1.97 | 22.87 | 57.87 | 2.80 | 22.20 | 62.80 | 5.68 | 22.03 | 58.72 |
| | Centroid | 4.85 | 23.71 | 62.81 | 3.43 | 23.66 | 66.18 | 3.48 | 22.80 | 67.18 | 3.78 | 23.71 | 62.33 |
| | LexRank | 5.12 | 24.04 | 68.36 | 5.09 | 22.49 | 57.91 | 5.09 | 25.22 | 84.44 | 4.49 | 22.19 | 58.11 |
| | QT | 3.81 | 22.32 | 59.32 | 3.49 | 22.80 | 65.18 | 2.13 | 22.38 | 73.91 | 5.04 | 22.19 | 66.43 |
| | SemAE | 0.41 | 22.07 | 55.64 | 4.55 | 21.81 | 53.13 | 5.05 | 21.82 | 63.59 | 0.61 | 21.61 | 56.39 |
| | HERCULES$_{\text{ext}}$ | 4.87 | 25.36 | 82.79 | 6.97 | 24.45 | 81.36 | 6.74 | 22.92 | 86.39 | 9.17 | 24.79 | 85.66 |
| *Abstractive* | CopyCat | 3.88 | 24.45 | 64.23 | 6.71 | 22.02 | 53.10 | 4.37 | 22.42 | 69.36 | 2.60 | 23.06 | 65.36 |
| | InstructGPT | 5.61 | 22.42 | 47.50 | 3.80 | 21.55 | 44.20 | 9.54 | 22.10 | 47.57 | 8.63 | 21.40 | 43.24 |
| | BiMeanVAE | 3.64 | 21.88 | 45.48 | 5.73 | 21.86 | 50.81 | 3.85 | 21.59 | 58.56 | 9.43 | 21.79 | 55.29 |
| | COOP | 5.61 | 22.95 | 53.23 | 6.76 | 22.74 | 63.97 | 2.73 | 22.14 | 60.76 | 9.46 | 22.19 | 55.45 |
| | HERCULES$_{\text{abs}}$ | 6.54 | 25.55 | 79.49 | 8.98 | 25.25 | 82.15 | 5.97 | 24.59 | 85.09 | 9.19 | 25.53 | 84.16 |
| | (References) | 87.80 | 87.69 | 63.72 | 87.87 | 87.11 | 65.12 | 92.32 | 85.49 | 69.86 | 89.63 | 86.73 | 67.58 |

Table 11: Results for automatic faithfulness metrics on AmaSum, broken down by product category.

| | | SPACE | | | AmaSum | | | Overall | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | System | Info ↑ | Cohe ↑ | Conc ↑ | Info ↑ | Cohe ↑ | Conc ↑ | Info ↑ | Cohe ↑ | Conc ↑ |
| *Extractive* | Random | -5.33 | -2.67 | -4.67 | -14.04 | 3.07 | -1.75 | -9.68 | 0.20 | -3.21 |
| | LexRank | -27.33 | -39.33 | -44.67 | 7.05 | -5.29 | -4.41 | -10.14 | -22.31 | -24.54 |
| | QT | -8.67 | -10.00 | -4.67 | -7.42 | -0.87 | 5.68 | -8.05 | -5.44 | 0.51 |
| | HERCULES$_{\text{ext}}$ | 1.33 | -2.00 | 0.00 | -1.54 | -1.98 | 5.05 | -0.10 | -1.99 | 2.53 |
| | (Gold) | 40.00 | 54.00 | 54.00 | 20.00 | 6.30 | -5.75 | 30.00 | 30.15 | 24.12 |
| *Abstractive* | Random | -18.67 | -2.67 | -11.33 | -22.67 | -6.22 | -1.78 | -20.67 | -4.44 | -6.56 |
| | InstructGPT | -10.67 | 5.33 | 18.00 | 11.11 | 2.22 | 0.89 | 0.22 | 3.78 | 9.44 |
| | COOP | -12.00 | -24.67 | -20.00 | -4.00 | -0.89 | 0.89 | -8.00 | -12.78 | -9.56 |
| | HERCULES$_{\text{abs}}$ | 4.67 | -16.00 | -18.67 | -1.78 | -8.44 | -5.33 | 1.44 | -12.22 | -12.00 |
| | (References) | 36.67 | 38.00 | 32.00 | 21.67 | 16.67 | 6.67 | 29.17 | 27.33 | 19.33 |

Table 12: Breakdown of human evaluation results by dataset.

| | |
|---|---|
| *Rating = 1 (bad)* | Then it stopped working. It died in less than a year. Do not buy this machine. It didn't even last a year. Bought this in January 2017. |
| *Rating = 5 (good)* | This is a great fan. It's very quiet. I love this fan. The light is bright. This is a very nice remote. |
| *Rating = 1 (bad)* | The carpet was stained and dirty. The room was filthy. The bathroom was disgusting. The staff was unfriendly and unhelpful. Avoid this hotel at all costs. |
| *Rating = 5 (good)* | The hotel is very close to the airport. The shuttle service was great. The pool and hot tub were great. The food was delicious. The view from our room was breathtaking. |

Table 13: Examples of sentiment-controlled summaries generated by HERCULES, from SPACE and AmaSum.

| System | Output |
|---|---|
| *Reference* | The staff were helpful and friendly, especially in the spa! The hotel is consistently clean and well kept and The rooms are luxurious. The rooms have also been upgraded with new white comforters and new ruffle dusters. The hotel also features kitchens so that we can prepare breakfast and eat out in the beautiful garden and lunch at the picnic area so that we can barbeque, but the food in wine country is also wonderful! The location of the hotel is within walking distance to all the shops and restaurants. The grounds are beautiful with the heated indoor/outdoor pools . |
| HERCULES$_{\text{ext}}$ | The staff was friendly and helpful. The pools well maintained. The rooms are clean and spacious. Thank you Roman Spa for being a great place to stay. It seems that the hotel is improving. The pools and hot tubs are fantastic. The grounds are very well maintained, with beautiful flowers, and waterfalls all throughout. Again, no service seems to be the mantra at the Roman Spa. The grounds were nice. The location is great and allows walking to all Calistoga spas, restaurants, and shopping. |
| HERCULES$_{\text{abs}}$ | The staff was very friendly and helpful. the pool was great. The rooms were clean and comfortable. We had a great time. The hotel is in a great location. the pool and hot tub were great. The grounds are beautiful and well maintained. service was impeccable. The grounds are very nice and well maintained. The hotel is in a great location, close to everything. |
| *Reference* | This uniquely shaped vacuum is ideal for sucking debris out of corners and tight spaces. V-shaped design drives debris in front of the vacuum for easier clean-up. Swivel head makes it easy to maneuver around furniture. Power cord extends 20 feet. Great for picking up debris in tight spots with no attachments needed. Runs quietly. Users say this vacuum is prone to clogging and requires routine maintenance |
| HERCULES$_{\text{ext}}$ | Works great on my Handscraped, hardwood floors. This vacuum is not made for carpet, but works well on any hard flat surface. Lightweight and easy to maneuver. There are three areas that trap dirt and need to be cleaned in order to empty the vacuum. We bought this little vacuum for our new hardwood floors, and it's the best thing ever. My only complaint is the cord is pretty short. I love this vacuum!! This vacuum is amazing. Love this little vacuum. The suction is great! It picks up pet hair and dirt as advertised. And it picks up dust!! I have 4 cats and 2 dogs. |
| HERCULES$_{\text{abs}}$ | The suction power is great. No scrubbing necessary. This little vacuum is amazing. Easy to maneuver. Great for hard floors. No more dust Bunnies! My dog sheds so much. This vacuum is amazing! Does not stay in place. Excellent customer service. The cord is too short. Love this vacuum! Great for pet hair. |
| *Reference* | A gaming-specific external hard drive designed for the whole range of Xbox consoles. Two terabyte options (2TB and 4TB) offer plenty of space for installing games, apps, and files. Can be used with multiple Xbox consoles. The noise level of the hard drive is louder than most other options |
| HERCULES$_{\text{ext}}$ | Very noticeably speeds up loading times for gaming on my Xbox. Tons of storage now for my Xbox one. No issues at all. All Seagate though. Used for Xbox one. It ' s just been sitting on my TV stand connected to me Xbox one X. Plug and play. So we ended up buying an external drive. Super easy to install. I have about 50 games installed on this hard drive and still have 75% of space left. Love this Ssd! Plenty of space for extra games. Stopped working after 2 years. This hard drive is awesome. |
| HERCULES$_{\text{abs}}$ | Stopped working in less than a year. Easy to install and use. Works great with my Macbook pro. Plug and play. Plenty of room. This one does. Fast load times. Great for gaming. This hard drive is very fast. No SD card reader. This external hard drive is great. This Ssd is fast. Bought this for my bedroom. Great price and fast shipping. |

Table 14: Additional examples of output from HERCULES from both SPACE and AmaSum.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Limitations section at end*

☑ A2. Did you discuss any potential risks of your work?
*Limitations section at end*

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

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

### B ☑ Did you use or create scientific artifacts?

*4.4*

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

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*License will be distributed on Github*

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

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Not applicable. Left blank.*

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Not applicable. Left blank.*

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

### C ☑ Did you run computational experiments?

*4.4*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*Appendix A*

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*4.4/Appendix A*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*5*

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

**D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?**
*5*

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

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

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

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*Not applicable. Left blank.*

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*Appendix B*