Unsupervised Abstractive Opinion Summarization by Generating Sentences with Tree-Structured Topic Guidance

Masaru Isonuma[1] **Junichiro Mori**[1,2] **Danushka Bollegala**[3] **Ichiro Sakata**[1]
[1]The University of Tokyo, Japan [2] RIKEN, Japan [3] University of Liverpool, United Kingdom
isonuma@ipr-ctr.t.u-tokyo.ac.jp mori@mi.u-tokyo.ac.jp
danushka@liverpool.ac.uk isakata@ipr-ctr.t.u-tokyo.ac.jp

Abstract

This paper presents a novel unsupervised abstractive summarization method for opinionated texts. While the basic variational autoencoder-based models assume a unimodal Gaussian prior for the latent code of sentences, we alternate it with a *recursive Gaussian mixture*, where each mixture component corresponds to the latent code of a topic sentence and is mixed by a tree-structured topic distribution. By decoding each Gaussian component, we generate sentences with *tree-structured topic guidance*, where the root sentence conveys generic content, and the leaf sentences describe specific topics. Experimental results demonstrate that the generated topic sentences are appropriate as a summary of opinionated texts, which are more informative and cover more input contents than those generated by the recent unsupervised summarization model (Bražinskas et al., 2020). Furthermore, we demonstrate that the variance of latent Gaussians represents the granularity of sentences, analogous to Gaussian word embedding (Vilnis and McCallum, 2015).

1 Introduction

Summarizing opinionated texts, such as product reviews and online posts on Web sites, has attracted considerable attention recently along with the development of e-commerce and social media. Although extractive approaches are widely used in document summarization (Erkan and Radev, 2004; Ganesan et al., 2010), they often fail to provide an overview of the documents, particularly for opinionated texts (Carenini et al., 2013; Gerani et al., 2014). Abstractive summarization can overcome this challenge by paraphrasing and generalizing an entire document. Although supervised approaches have seen significant success with the development of neural architectures (See et al., 2017; Fabbri et al., 2019), they are limited to specific domains, e.g., news articles, where a large number of gold summaries are available. However, the domain of opinionated texts is diverse; manually writing gold summaries is therefore costly.

This lack in gold summaries has motivated prior work to develop unsupervised abstractive summarization of opinionated texts, for example, product reviews (Chu and Liu, 2019; Bražinskas et al., 2020; Amplayo and Lapata, 2020). While they generated consensus opinions by condensing input reviews, two key components were absent: *topics* and *granularity* (i.e., the level of detail). For instance, as shown in Figure 1, a gold summary of a restaurant review provides the overall impression and details about certain topics, such as food, ambience, and service. Hence, a summary typically comprises diverse topics, some of which are described in detail, whereas others are mentioned concisely.

From this investigation, we capture the *topic-tree structure* of reviews and generate *topic sentences*, that is, sentences summarizing specified topics. In the topic-tree structure, the root sentence conveys generic content, and the leaf sentences mention specific topics. From the generated topic sentences, we extract sentences with appropriate topics and levels of granularity as a summary. Regarding extractive summarization, capturing topics (Titov and McDonald, 2008; Isonuma et al., 2017; Angelidis and Lapata, 2018) and topic-tree structure (Celikyilmaz and Hakkani-Tur, 2010, 2011) is useful for detecting salient sentences. To the best of our knowledge, this is the first study to use the topic-tree structure in unsupervised abstractive summarization.

The difficulty of generating sentences with tree-structured topic guidance lies in controlling the granularity of topic sentences. Wang et al. (2019) generated a sentence with designated topic guidance, assuming that the latent code of an input sentence can be represented by a Gaussian mixture

Transactions of the Association for Computational Linguistics, vol. 9, pp. 945–961, 2021. https://doi.org/10.1162/tacl_a_00406
Action Editor: Asli Celikyilmaz. Submission batch: 3/2021; Revision batch: 4/2021; Published 9/2021.

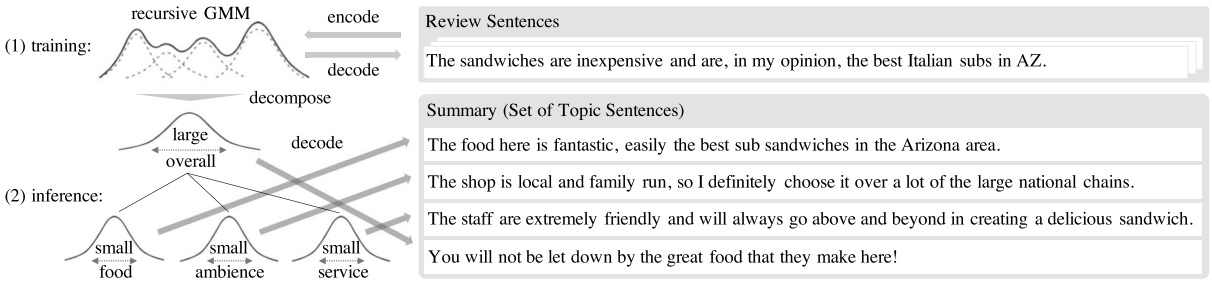


Figure 1: Outline of our approach. (1) The latent distribution of review sentences is represented as a recursive GMM and trained in an autoencoding manner. Then, (2) the topic sentences are inferred by decoding each Gaussian component. An example of a restaurant review and its corresponding gold summary are displayed.

model (**GMM**), where each Gaussian component corresponds to the latent code of a topic sentence. While they successfully generated a sentence relating to a designated topic by decoding each mixture component, modelling the sentence granularity in a latent space to generate topic sentences with multiple granularities remains to be realized.

To overcome this challenge, we model the sentence granularity by the variance size of the latent code. We assume that general sentences have more uncertainty and are generated from a latent distribution with a larger variance, analogous to Gaussian word embedding (Vilnis and McCallum, 2015). Based on this assumption, we represent the latent code of topic sentences with Gaussian distributions, where the parent Gaussian receives a larger variance and represents a more generic topic sentence than its children, as shown in Figure 1. To obtain the latent code characterized above, we introduce a *recursive Gaussian mixture* prior to modeling the latent code of input sentences in reviews. A recursive GMM consists of Gaussian components that correspond to the nodes of the topic-tree, and the child priors are set to the inferred parent posterior. Because of this configuration, the Gaussian distribution of higher topics receives a larger variance and conveys more general content than lower topics.

The contributions of our work are as follows:

- We propose a novel unsupervised abstractive opinion summarization method by generating sentences with tree-structured topic guidance.
- To model the sentence granularity in a latent space, we specify a Gaussian distribution as the latent code of a sentence and demonstrate that the granularity depends on the variance size.
- Experiments demonstrate that the generated summaries are more informative and cover more input content than the recent unsupervised summarization (Bražinskas et al., 2020).

2 Preliminaries

Bowman et al. (2016) adapted the variational autoencoder (**VAE**; Kingma and Welling, 2014; Rezende et al., 2014) to obtain the density-based latent code of sentences. They assume the generative process of documents to be as follows:

For each document index $d \in \{1, \ldots, D\}$:
For each sentence index $s \in \{1, \ldots, S_d\}$ in d:

1. Draw a latent code of the sentence $\boldsymbol{x}_s \in \mathcal{R}^n$:

$$\boldsymbol{x}_s \sim p(\boldsymbol{x}_s) \tag{1}$$

2. Draw a sentence $\boldsymbol{w}_s$:

$$\boldsymbol{w}_s | \boldsymbol{x}_s \sim p(\boldsymbol{w}_s | \boldsymbol{x}_s) = \mathrm{RNN}(\boldsymbol{x}_s) \tag{2}$$

where $p(\boldsymbol{w}_s|\boldsymbol{x}_s) = \prod_t p(w_s^t | \boldsymbol{w}_s^{<t}, \boldsymbol{x}_s)$ is derived by an recurrent neural networks (**RNN**) decoder. The latent prior is a standard Gaussian: $p(\boldsymbol{x}_s) = \mathcal{N}(\boldsymbol{x}_s | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$. The likelihood of a document and its evidence lower bound (**ELBO**) are given by (3) and (4), respectively:

$$p(\boldsymbol{W}_{1:S_d}) = \prod_{s=1}^{S_d} \left\{ \int p(\boldsymbol{w}_s|\boldsymbol{x}_s) p(\boldsymbol{x}_s) d\boldsymbol{x}_s \right\} \tag{3}$$

$$\mathcal{L}_d = \sum_{s=1}^{S_d} \left\{ \mathbf{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)} \left[\log p(\boldsymbol{w}_s|\boldsymbol{x}_s) \right] - \mathrm{D}_{\mathrm{KL}} \left[q(\boldsymbol{x}_s|\boldsymbol{w}_s) | p(\boldsymbol{x}_s) \right] \right\} \tag{4}$$

$q(\boldsymbol{x}_s|\boldsymbol{w}_s) = \mathcal{N}(\boldsymbol{x}_s | \hat{\boldsymbol{\mu}}_s, \hat{\boldsymbol{\Sigma}}_s)$ is the variational distribution with $\hat{\boldsymbol{\mu}}_s = f_\mu(\boldsymbol{w}_s), \hat{\boldsymbol{\Sigma}}_s = \mathrm{diag}[f_\Sigma(\boldsymbol{w}_s)]$ where f_μ and f_Σ are RNN encoders.

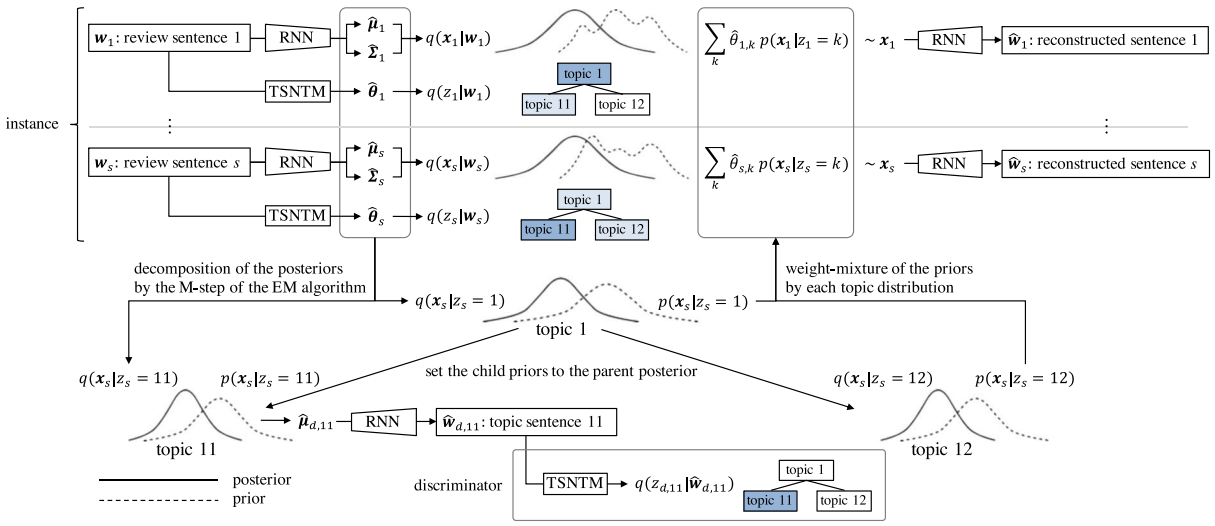


Figure 2: Outline of our model. We set a recursive Gaussian mixture as the latent prior of review sentences and obtain the latent posteriors of topic sentences by decomposing the posteriors of review sentences.

By representing sentences by Gaussians rather than vectors, the decoded sentence from the intermediate latent code between two sentences is grammatical and has a coherent topic with the two sentences. Extending their work, we construct the prior as a recursive GMM and infer the topic sentences by decoding each Gaussian component.

3 RecurSum: Recursive Summarization

In this section, we explain our model, RecurSum. Figure 2 shows the outline. The latent code of review sentences is obtained as a recursive GMM (3.1), and topic sentences are inferred by decoding each Gaussian component (3.2). A summary is then created by extracting the appropriate topic sentences (3.3). We introduce additional components to improve the quality of topic sentences (3.4) and explain why general/specific content is conveyed by the root/leaf topics, referring to the analogy with Gaussian word embedding (3.5).

3.1 Generative Model of Reviews

We assume the generative process of reviews to be as follows. We refer to the set of sentences in multiple reviews of a specific product as *instance*. Compared to Bowman et al. (2016), we explicitly model the topic of review sentences as follows:

For each instance index $d \in \{1, \ldots, D\}$:

For each sentence index $s \in \{1, \ldots, S_d\}$ in d:

1. Draw a topic of the sentence $z_s \in \{1, \ldots, K\}$:

$$z_s \sim \text{Mult}(\boldsymbol{\theta}) \tag{5}$$

2. Draw a latent code of the sentence $\boldsymbol{x}_s \in \mathcal{R}^n$:

$$\boldsymbol{x}_s | z_s \sim \prod_{k=1}^{K} p(\boldsymbol{x}_s | z_s = k)^{\delta(z_s = k)} \tag{6}$$

3. Draw a review sentence $\boldsymbol{w}_s$:

$$\boldsymbol{w}_s | \boldsymbol{x}_s \sim p(\boldsymbol{w}_s | \boldsymbol{x}_s) = \text{RNN}(\boldsymbol{x}_s) \tag{7}$$

where the topic distribution is tree-structured, and its prior is set to be uniform. In (6), we assume a recursive GMM as the latent prior of a review sentence (δ is a Dirac delta). Each mixture component corresponds to the latent distribution of a sentence conditioned on a specific topic, $p(\boldsymbol{x}_s | z_s = k)$:

$$p(\boldsymbol{x}_s | z_s = 1) = \mathcal{N}(\boldsymbol{x}_s | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \tag{8}$$

$$\begin{aligned} p(\boldsymbol{x}_s | z_s = k) &= q(\boldsymbol{x}_s | z_s = par(k)) \\ &= \mathcal{N}(\boldsymbol{x}_s | \hat{\boldsymbol{\mu}}_{d,par(k)}, \hat{\boldsymbol{\Sigma}}_{d,par(k)}) \ (k \neq 1) \end{aligned} \tag{9}$$

where $par(k)$ denotes the parent of the k-th topic. $q(\boldsymbol{x}_s | z_s = par(k))$ is the approximated latent posterior of the parent topic sentence as derived later in Section 3.2. We assume that the latent posterior of the parent sentence is appropriate as the latent prior of its child sentences.

Under our generative model, the likelihood of an instance and its ELBO are given by (10) and (11), respectively:

$$p(\boldsymbol{W}_{1:S_d}) = \prod_{s=1}^{S_d} \int p(\boldsymbol{w}_s | \boldsymbol{x}_s) p(\boldsymbol{x}_s | z_s) p(z_s) d\boldsymbol{x}_s dz_s \tag{10}$$

$$
\begin{aligned}
\mathcal{L}_d = & \sum_{s=1}^{S_d} \Big\{ \mathbf{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)} \big[\log p(\boldsymbol{w}_s|\boldsymbol{x}_s)\big] \\
& - \mathbf{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)q(z_s|\boldsymbol{w}_s)} \big[\log q(\boldsymbol{x}_s|\boldsymbol{w}_s) - \log p(\boldsymbol{x}_s|z_s)\big] \\
& - \mathbf{E}_{q(z_s|\boldsymbol{w}_s)} \big[\log q(z_s|\boldsymbol{w}_s) - \log p(z_s)\big] \Big\} \\
= & \sum_{s=1}^{S_d} \Big\{ \mathbf{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)} \big[\log p(\boldsymbol{w}_s|\boldsymbol{x}_s)\big] - \mathrm{D}_{\mathrm{KL}} \big[q(z_s|\boldsymbol{w}_s)|p(z_s)\big] \Big\} \\
& - \sum_{k=1}^{K} \sum_{s=1}^{S_d} \Big\{ \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}} \big[q(\boldsymbol{x}_s|\boldsymbol{w}_s)|p(\boldsymbol{x}_s|z_s=k)\big] \Big\}
\end{aligned}
\tag{11}
$$

where $q(\boldsymbol{x}_s|\boldsymbol{w}_s) = \mathcal{N}(\boldsymbol{x}_s|\hat{\boldsymbol{\mu}}_s, \hat{\boldsymbol{\Sigma}}_s)$ is the latent posterior of a sentence s, inferred by an RNN encoder. $\hat{\theta}_{s,k} = q(z_s = k|\boldsymbol{w}_s)$ is the variational topic distribution and inferred by the tree-structured neural topic model (**TSNTM**; Isonuma et al., 2020). More details are provided in Appendix A.1.

3.2 Inference of Topic Sentences

From the latent posterior of review sentences, we infer the latent posterior of each topic sentence using the M-step of the EM algorithm. We define the variational distribution of the latent code of a topic sentence as (12) and compute the Gaussian parameters as (13) and (14) that maximize $\sum_{s=1}^{S_d} \mathbf{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)q(z_s|\boldsymbol{w}_s)} \big[\log q(\boldsymbol{x}_s|z_s)\big]$ as follows:

$$
q(\boldsymbol{x}_s|z_s) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_s|\hat{\boldsymbol{\mu}}_{d,k}, \hat{\boldsymbol{\Sigma}}_{d,k})^{\delta(z_s=k)}
\tag{12}
$$

$$
\hat{\boldsymbol{\mu}}_{d,k} = \frac{\sum_{s=1}^{S_d} \hat{\theta}_{s,k} \mathbf{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)}[\boldsymbol{x}_s]}{\sum_{s=1}^{S_d} \hat{\theta}_{s,k}} = \frac{\sum_{s=1}^{S_d} \hat{\theta}_{s,k} \hat{\boldsymbol{\mu}}_s}{\sum_{s=1}^{S_d} \hat{\theta}_{s,k}}
\tag{13}
$$

$$
\begin{aligned}
\hat{\boldsymbol{\Sigma}}_{d,k} &= \frac{\sum_{s=1}^{S_d} \hat{\theta}_{s,k} \mathbf{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)}[(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})^\top]}{\sum_{s=1}^{S_d} \hat{\theta}_{s,k}} \\
&= \frac{\sum_{s=1}^{S_d} \hat{\theta}_{s,k} \{\hat{\boldsymbol{\Sigma}}_s + (\hat{\boldsymbol{\mu}}_s - \hat{\boldsymbol{\mu}}_{d,k})(\hat{\boldsymbol{\mu}}_s - \hat{\boldsymbol{\mu}}_{d,k})^\top\}}{\sum_{s=1}^{S_d} \hat{\theta}_{s,k}}
\end{aligned}
\tag{14}
$$

From these latent posteriors, we generate the topic sentences for each instance using the respective mean rather than a sample: $\hat{\boldsymbol{w}}_{d,k} \sim p(\boldsymbol{w}_{d,k}|\hat{\boldsymbol{\mu}}_{d,k}) = \mathrm{RNN}(\hat{\boldsymbol{\mu}}_{d,k})$. Similar to Bražinskas et al. (2020); Chu and Liu (2019), we assume that the average latent code represents the common contents of the corresponding topic, while specific contents are distributed apart from the mean. Therefore, decoding the mean rather than a sample would be desirable for generating a summary.

3.3 Extraction of Summary Sentences

Next, we create a summary by extracting appropriate sentences from the generated topic sentences. As gold summaries are not available for training, we need a measure to evaluate candidate summaries using only input reviews. As reported in Chu and Liu (2019), the ROUGE scores (Lin, 2004) between a candidate summary and the *input reviews* effectively measure the extent to which the summary encapsulates the reviews. Based on this assumption, we search the topic sentences by maximizing the ROUGE-1 F-measure with the review sentences in an instance. We use a beam search and keep multiple highest-score candidates for each step. Similar to Carbonell and Goldstein (1998), to eliminate the redundancy of summary sentences, we do not add a sentence with a high word overlap (ROUGE-1 precision) against the sentences already included in the summary. The hyperparameters are tuned based on the validation set, as described in Section 4.2.

After selecting the summary sentences, we sort them in the depth-first order according to the topic-tree structure—that is, we begin at the root node and explore as far as possible along each branch before backtracking. Barzilay and Lapata (2008) advocate that adjacent sentences in the coherent text tend to have similar contents. As we assume that sentences linked by parent-child relations are topically coherent, the generated summary is expected to be locally coherent by extracting child sentences after their parent sentence.

3.4 Additional Model Components

The basic components of our model have been explained in the previous sections. This section introduces three additional components to improve the quality of topic sentences. In ablation studies (Section 5.2), we will see the effect of these components on summarization performance.

Discriminator To ensure that each topic sentence has a specific topic, we introduce a discriminator following Hu et al. (2017) and Tang et al. (2019). We approximate the sample of the topic sentence by using the Gumbel-softmax trick (Jang et al., 2017; Maddison et al., 2017) and reuse the TSNTM to estimate the topic distribution of the sample, $q(z_{d,k}|\hat{\boldsymbol{w}}_{d,k})$. By maximizing the likelihood of the specified topic as (15), the discriminator forces the generated k-th topic sentence to be coherent with topic k.

$$
\mathcal{L}_d^{disc} = \sum_{k=1}^{K} \log q(z_{d,k} = k|\hat{\boldsymbol{w}}_{d,k})
\tag{15}
$$

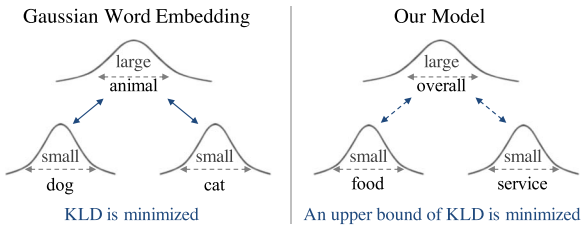


Figure 3: Analogy with Gaussian word embedding.

Attention We use the attention-based RNN decoder (Luong et al., 2015) to efficiently reflect input sentence information into output topic sentences. Given the hidden state of the t-th word in an output sentence $\boldsymbol{h}_o^t$ and the i-th word in an input review sentence $\boldsymbol{h}_s^i$, we calculate the attention distribution over all the words in the input review sentences to compute the word probability.

$$a(\boldsymbol{h}_o^t, \boldsymbol{h}_s^i) = \frac{\exp(\boldsymbol{h}_o^{t\top} \boldsymbol{h}_s^i)}{\sum_{s'} \sum_{i'} \exp(\boldsymbol{h}_o^{t\top} \boldsymbol{h}_{s'}^{i'})} \quad (16)$$

$$\boldsymbol{c}_o^t = \sum_{s'} \sum_{i'} a(\boldsymbol{h}_o^t, \boldsymbol{h}_{s'}^{i'}) \boldsymbol{h}_{s'}^{i'} \quad (17)$$

$$p(w_o^t | \boldsymbol{w}_o^{<t}, \hat{\boldsymbol{\mu}}_o) = \text{softmax}(\boldsymbol{W}[\boldsymbol{h}_o^t; \boldsymbol{c}_o^t]) \quad (18)$$

Nucleus Sampling During the inference, we use nucleus sampling (Holtzman et al., 2019) to decode the topic sentences. Holtzman et al. (2019) reported that maximization-based decoding methods such as beam search tend to generate bland, incoherent, and repetitive text in open-ended text generation. As we will see in the ablation experiments, nucleus sampling is effective in generating diverse and informative topic sentences.

3.5 Analogy with Gaussian Word Embedding

Here, we explain why a general sentence is generated from the root topic, while more specific content is conveyed by the sentences generated by the leaf topics, referring to Gaussian word embedding.

Gaussian word embedding (Vilnis and McCallum, 2015) represents words as Gaussian distributions and captures the hierarchical relations among the words. As shown in Figure 3, by representing words as densities over a latent space and minimizing the KL-divergence of the distributions, they detect that common words such as ‘‘animal’’ obtain a larger variance than more specific words, such as ‘‘dog’’ and ‘‘cat’’. This can be explained by the fact that general words have more uncertainty in their meaning (i.e.,‘‘animal’’ sometimes denotes ‘‘dog’’ and other times ‘‘cat’’).

Similarly, our model minimizes the upper bound of the KL-divergence of the latent distribution between a parent topic sentence and its children. In (19), we show that the x-related term in the ELBO (11) is an upper bound of the KL-divergence of the latent posteriors between parent-child topic sentences (derived in Appendix A.3).

$$\begin{aligned}
&\textstyle\sum_{s=1}^{S_d} \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\left[q(\boldsymbol{x}_s|\boldsymbol{w}_s)|p(\boldsymbol{x}_s|z_s=k)\right] \\
&\geq \textstyle\sum_{s=1}^{S_d} \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\left[q(\boldsymbol{x}_s|z_s=k)|p(\boldsymbol{x}_s|z_s=k)\right] \\
&= \textstyle\sum_{s=1}^{S_d} \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\left[q(\boldsymbol{x}_s|z_s=k)|q(\boldsymbol{x}_s|z_s=par(k))\right]
\end{aligned} \quad (19)$$

since $p(\boldsymbol{x}_s|z_s=k) = q(\boldsymbol{x}_s|z_s=par(k))$ as defined in (9). Similar to Gaussian word embedding, maximizing the ELBO forces the latent distribution of a parent to be close to that of its children, and the parent receives a larger variance than its children. This property ensures that the parent-child topics have a coherent topic, and more general content is conveyed by the root topic sentences. Intuitively, a general sentence, such as ‘‘I love this restaurant’’, includes several topics, such as ‘‘food’’ and ‘‘service’’, and has a large uncertainty of semantics. Thus, we assume that a generic sentence is represented by the mean of the latent distribution with a larger variance, whereas a more specific sentence is generated from the distribution with a smaller variance.

Similar to Vilnis and McCallum (2015), we observed that the eigenvalues of the full covariance of topic sentences (14) become extremely small during training. To maintain a reasonably sized and positive semi-definite covariance, we add a hard constraint to the diagonal covariance of the review sentences as $\hat{\boldsymbol{\Sigma}}_{s,ii} \leftarrow \max(\lambda, \hat{\boldsymbol{\Sigma}}_{s,ii})$ since $\log|\hat{\boldsymbol{\Sigma}}_{d,k}| \geq (\sum_s \hat{\theta}_{s,k} \log|\hat{\boldsymbol{\Sigma}}_s|)/(\sum_s \hat{\theta}_{s,k}) \geq n \log \lambda$, as derived in Appendix A.3.

4 Experiments

4.1 Datasets

In our experiments, we used the ***Yelp*** *Dataset Challenge*[1] and ***Amazon*** *product reviews* (McAuley et al., 2015). By pre-processing the reviews similarly as in Chu and Liu (2019) and Bražinskas et al. (2020), we obtained the dataset as shown in Table 1. Regarding the training set,

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

Dataset	Yelp	Amazon
Training	173,088	280,692
Validation	100	84
Test	100	96

Table 1: Number of instances (pairs of eight reviews and a gold summary) in the datasets. The training set does not contain gold summaries.

we removed products[2] with fewer than 8 reviews and reviews in which the maximum number of sentences exceeds 50. To prevent the dataset from being dominated by a small number of products, we created 12 and 2 instances for each product in Yelp and Amazon, respectively. Then, we randomly selected 8 reviews to construct an instance. Regarding the validation/test set of Yelp, we randomly split 200 instances provided by Chu and Liu (2019)[3] into validation and test sets. For Amazon, we used the same validation and test sets provided by Bražinskas et al. (2020).[4] These gold summaries were created by Amazon Mechanical Turk (**AMT**) workers, who summarized 8 reviews for each product. The vocabulary comprises words that appear more than 16 times in the training set. The vocabulary sizes are 31,748 and 30,732 for Yelp and Amazon, respectively.

4.2 Implementation Details

We set the hyperparameters as follows, which maximize the ROUGE-L in the validation set of Yelp and use the same hyperparameters on-Amazon.[5] The dimensions of word embeddings and the latent code of the sentences are 200 and 32, respectively. The encoder and decoder are single-layer bi-directional and uni-directional GRU-RNN (Chung et al., 2014) with 200-dimensional hidden units for each direction. The threshold of nucleus sampling is 0.4. We train our model using Adam (Kingma and Ba, 2014) with a learning rate of 5.0×10^{-3}, a batch size of 8, and a dropout rate of 0.2. The initial Gumbel-softmax temperature is set to 1 and decreased by 2.5×10^{-5} per training step. Similar to Bowman et al. (2016) and Yang et al. (2017), we avoid posterior collapse by increasing the weight of the KL-term by 2.5×10^{-5} per training step. We set the review sentence's minimum covariance to $\lambda = \exp(0.5)$. Regarding the tree structure, we set the number of levels to 3, and the number of branches to 4 for both the second and third levels. The total number of topics is 21. Regarding the summary sentence extractor in Section 3.3, we set the maximum number of extracted sentences as 6, the beam width as 8, and the redundancy threshold as 0.6.

4.3 Baseline Methods

As a baseline, we use *Multi-Lead-1*, which extracts the first sentence of each review. Furthermore, we employ unsupervised extractive approaches, *LexRank* (Erkan and Radev, 2004) and *Opinosis* (Ganesan et al., 2010). LexRank is a PageRank-based sentence extraction method that constructs a graph in which sentences and their similarity are represented by the nodes and edges, respectively. Opinosis constructs a word-based graph and extracts redundant phrases as a summary. As unsupervised abstractive summarization methods, we use *MeanSum* (Chu and Liu, 2019), *Copycat* (Bražinskas et al., 2020), and *DenoiseSum* (Amplayo and Lapata, 2020). MeanSum computes the mean of the review embeddings and decodes it as a summary. Copycat generates a consensus opinion by a hierarchical VAE which is trained by generating a new review given a set of other reviews of a product. DenoiseSum[6] creates synthetic reviews by adding noise to original reviews and generates a summary by removing non-salient information as noise.

As an upper bound of extraction methods, we also report the performance of *Oracle*, which extracts the topic sentences such that they obtain the highest ROUGE-L against each gold summary. As the average number of sentences in the gold summaries is approximately four, we extract four topic sentences to generate a summary.

4.4 Semi-automatic Evaluation of Summaries

Following Chu and Liu (2019) and Bražinskas et al., 2020, we use the ROUGE-1/2/L F1-scores (Lin, 2004) as semi-automatic evaluation metrics.

Table 2 shows the rouge scores of our model, *RecurSum*, and the baselines for the test sets. In

[2]We refer to businesses (e.g., a specific Starbucks branch) in Yelp and products (e.g., iPhone X) in Amazon as *products*.

[3]https://github.com/sosuperic/MeanSum.

[4]https://github.com/abrazinskas/Copycat-abstractive-opinion-summarizer.

[5]https://github.com/misonuma/recursum.

[6]As complete code is not available, we report the result of different test splits from ours, which are used in their sample of output summaries. https://github.com/rktamplayo/DenoiseSum.

	Yelp			Amazon		
Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Multi-Lead-1	27.42	3.74	14.34	30.32	5.85	15.96
LexRank (Erkan and Radev, 2004)	26.40	3.19	14.35	31.42	5.31	16.70
Opinosis (Ganesan et al., 2010)	25.80	2.92	14.57	28.90	4.11	16.33
MeanSum (Chu and Liu, 2019)	28.66	3.73	15.77	30.16	4.51	17.76
Copycat (Bražinskas et al., 2020)	28.95	4.80	17.76	31.84	5.79	**20.00**
DenoiseSum (Amplayo and Lapata, 2020)	29.77	5.02	17.63	–	–	–
RecurSum (Our Model)	**33.24**	**5.15**	**18.01**	**34.91**	**6.33**	18.91
RecurSum (Oracle)	35.59	7.93	28.63	37.17	9.85	30.19

Table 2: ROUGE F1-scores of the test set (%). Boldface shows the highest score excluding the oracle, and underlined scores are not regarded as statistically significant ($p < 0.05$) by approximate randomization test as compared to the highest score.

	Yelp				Amazon			
Model	Fluency	Coherence	Informative.	Redundancy	Fluency	Coherence	Informative.	Redundancy
LexRank	−16.88	−13.51	−0.64	−6.83	−18.18	−15.07	14.11	−4.76
MeanSum	5.63	−16.18	−13.73	0.70	2.74	−14.69	−13.70	1.32
Copycat	**15.07**	7.88	−7.19	**4.00**	**14.65**	9.80	−17.65	**6.85**
RecurSum	−2.56	**19.46**	**24.44**	2.78	0.70	**17.72**	**17.39**	−2.99

Table 3: Human evaluation scores on the *quality* of the summaries. The scores are computed by using the best-worst scaling (%) and range from −100 (unanimously worst) to +100 (unanimously best). Boldface denotes the highest score, and underlined scores are not regarded as statistically significant ($p < 0.05$) by Tukey HSD test as compared to the highest score.

most metrics on both datasets, our model outperforms MeanSum and achieves competitive performance compared with the recent unsupervised summarization model, Copycat. Regarding the oracle, our model significantly outperforms the other models. This result suggests that our model can improve the performance by using more sophisticated extraction methods. Although we have also attempted to use the integer linear programming-based method (Gillick and Favre, 2009), it did not improve the performance. Developing such extraction techniques is beyond the scope of the current study, which focuses on topic structure, and is deferred to future work.

4.5 Human Evaluation of Summaries

We conducted a human evaluation using AMT. Following Bražinskas et al. (2020) and Amplayo and Lapata (2020), we randomly selected 50 instances from each test set and asked AMT workers[7] to complete the following three tasks:

Quality of the Summaries We presented four system summaries in random order and asked six AMT workers to rank the summzarization quality referring to the gold summary. We compute each system's score as the percentage of times selected as the best minus those are selected as the worst by using the best-worst scaling (Louviere et al., 2015; Kiritchenko and Mohammad, 2016).

Following Bražinskas et al. (2020) and Amplayo and Lapata (2020), we use the following four criteria: *Fluency*: the summary is grammatically correct, easy to read, and understand; *Coherence*: the summary is well structured and organized; *Informativeness*: the summary mentions specific aspects of the product; and *Redundancy*: the summary has no unnecessary repetitive words or phrases.

Table 3 shows the human evaluation scores of four systems. In terms of coherence and informativeness, RecurSum achieves the highest score among all approaches across the two datasets. This result indicates the effectiveness of considering topics and structure in unsupervised abstractive opinion summarization. With regard to fluency, Copycat is superior to our model because our

[7]To obtain reliable answers, we set the worker requirements to 98% approval rate, 1000+ accepted tasks, and locations in the US, UK, Canada, Australia, and New Zealand.

	Yelp		Amazon	
	Copycat	*RecurSum*	Copycat	*RecurSum*
Full	47.79	47.43	45.64	44.74
Partial	41.59	40.00	40.94	38.95
No	10.62	12.57	13.42	16.32

Table 4: Human evaluation scores on the *faithfulness* of the summaries (%). The difference of each system's frequency distribution is not regarded as statistically significant ($p < 0.05$) by χ^2 test.

	Yelp			Amazon		
	Copycat	*RecurSum*	Gold	Copycat	*RecurSum*	Gold
Full	23.94	**31.05**	34.52	29.15	**33.31**	38.41
Partial	30.02	**37.73**	40.91	29.15	**36.53**	39.63
No	46.04	**31.22**	24.58	41.71	**30.16**	21.96

Table 5: Human evaluation scores on the *coverage* of the summaries (%). The difference of the frequency distribution between Copycat and RecurSum is statistically significant ($p < 0.05$) by χ^2 test.

model sometimes makes a grammatical or referential error, which has a negative impact on fluency, as will be shown later in Section 5.1.

Faithfulness of the Summaries Abstractive summarization sometimes invents content that is unfaithful to the input texts (Maynez et al., 2020). The next study assesses whether the contents mentioned in the generated summaries are included in the input reviews. We use the same summary sets as in the quality evaluation and split them into sentences. For each summary sentence, we asked the AMT workers to judge whether the content is fully mentioned (Full), some of the content is mentioned (Partial), or no content is mentioned (No) in the reviews.

Table 4 shows the percentage of each answer. The frequency distribution is not regarded as statistically significant by the χ^2 test ($p < 0.05$). This result indicates that our model correctly reflects the content in the input reviews as well as Copycat.

Coverage of the Summaries Another desirable property of summaries is that they cover more content mentioned in the input reviews. As reported in Bražinskas et al. (2020), Copycat and MeanSum achieve relatively low scores for the human evaluation of *opinion consensus*, which captures the coverage of common opinions in the input reviews. In contrast, as RecurSum explicitly generates summary sentences for each topic, it could cover more input content across diverse topics. To assess this assumption, we conducted the opposite study from the faithfulness evaluation. Similar to the faithfulness evaluation, we split reviews into sentences. For each review sentence, we asked the AMT workers to rate the extent to which the generated summaries cover the input content.

Table 5 shows the percentage of fully-covered (Full), partially-covered (Partial), and un-covered (No) sentences. In addition to the two models, we also included gold summaries as the upper bounds. For both datasets, RecurSum covers more number of common opinions by capturing diverse topics.

5 Discussion

5.1 Analyzing Generated Summaries

In this section, we discuss the strengths and weaknesses of our method by presenting examples of the generated summaries and tree structures.

In Figure 4 (a), we present a summary of a review of shoes in Amazon. RecurSum generates topic sentences about fitness and size (12, 121), similar to Copycat. In addition, our model also mentions color and use (11, 111, 112), which is also described in the gold summary. While we cannot grasp that the shoes are appropriate for weddings from Copycat's summary, RecurSum covers such topics and provides more useful information.

Figure 4 (b) shows the generated summaries on a coffee shop review in Yelp. While both RecurSum and Copycat present a positive review about the taste of bubble tea (tea with tapioca), RecurSum also focuses on the dessert (12, 121), similar to the gold summary. While Copycat also refers to friendly staff, they are not mentioned in the input review. Our model successfully does not extract topic sentences about staff by measuring content overlap with the input reviews. However, RecurSum sometimes makes grammatical or referential errors such as ''*It's a little bit of the best bubble tea*''. These errors cause the inferior performance of RecurSum in terms of fluency.

Figure 4 (c) shows the summary of an Amazon review on a table chair set. RecurSum accurately captures opinions about the table (11, 111, 112) and chair (12, 121). The topic sentences on the

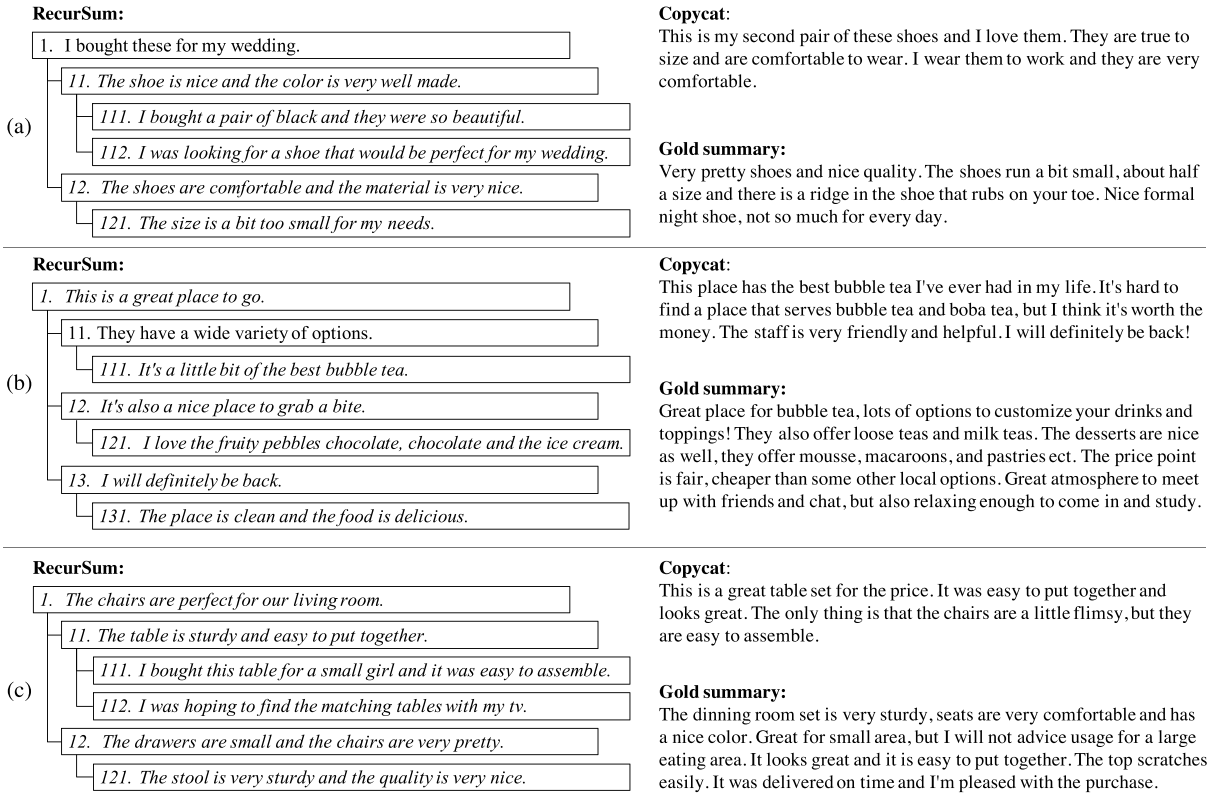


Figure 4: Generated topic sentences of (a) an Amazon review of heeled shoes, (b) a Yelp review of a coffee shop, and (c) an Amazon review of table chair set. Topic sentences selected as a summary are highlighted in *italic*.

bottom level elaborate on the parent sentences, referring to the easy assembly (111), the appropriate use of table (112), and the quality of chair (121). By inferring topics in the tree structure, RecurSum can offer summary sentences over multiple granularities of topics.

5.2 Ablation Study of Model Components

We report the results of the ablation study to investigate how individual components affect summarization performance. In addition to the ROUGE scores, we also report self-BLEU scores (Zhu et al., 2018) to investigate the diversity of the generated summaries. Self-BLEU is computed by calculating the BLEU score of each generated summary with all other generated summaries in the test set as references. A higher self-BLEU implies that the generated summaries are not diversified, that is, the model tends to generate a generic summary similar to the other summaries. Table 6 shows the performances of model variants on Yelp dataset.

w/o Disc denotes our model without a discriminator. The ROUGE scores are significantly lower than the full model. Without the discriminator, the topic distribution becomes sparse (i.e., most of the review sentences are assigned to some specific topics). Therefore, the model obtains incoherent topics and generates unfaithful summaries for the input review. Discriminator penalizes this situation by assigning an appropriate topic to topically different sentences. This mechanism makes the generated topic sentences topically coherent and improves ROUGE scores.

Model Variants	R-1	R-2	R-L	B-3	B-4
w/o Discriminator	30.52	3.50	16.43	54.18	30.42
w/o Attention	30.62	4.87	17.01	66.11	50.89
w/o Nucleus	31.71	5.10	17.70	69.13	55.81
Full	33.24	5.15	18.01	64.30	48.37

Table 6: Ablation study of RecurSum on Yelp. R-1/2/L denote ROUGE-1/2/L, respectively. B-3/4 denote self-BLEU3/4, respectively.

w/o Attention indicates our model without an attention mechanism. Although the generated sentences are faithful to the input review, they are often generic and miss some specific details of the content. By adding the attention mechanism, the

generated summary effectively reflects the content of the input reviews and provides more detailed information. Although the copy-mechanism (See et al., 2017) has also been reported to be useful in previous summarization models (Bražinskas et al., 2020; Amplayo and Lapata, 2020), it degrades the performance of our model. While their models use different input-output pairs (reviews vs. pseudo-summary), our model uses the same input-output pairs in an autoencoder manner and tends to fully copy the input sentences. Thus, our model fails to obtain a meaningful latent code.

w/o Nucleus denotes our model using a beam-search decoder (beam width = 5) instead of nucleus sampling when decoding topic sentences in inference. As reported by Holtzman et al. (2019), we also confirmed that the beam-search decoder tends to generate bland or repetitive text and sometimes fails to capture product-specific words. Owing to nucleus sampling, the decoder generates more informative content and improves the ROUGE-1 score with a significant decrease in self-BLEU.

We also attempted to replace the encoder with BERT (Devlin et al., 2019). However, fine-tuning of pretrained components with non-pretrained components is unstable as reported by Liu and Lapata (2019b), and it does not contribute to the improvement of ROUGE scores.

5.3 Analyzing Topic-Tree Structure

As generating sentences with tree-structured topic guidance is a novel challenge, we introduce new measures to verify that the generated sentences exhibit the desired properties of tree structures. Based on the work of tree-structured topic model (Kim et al., 2012), we introduce two metrics: *hierarchical affinity* and *topic specialization*.

Hierarchical Affinity. An important characteristic of the tree structure is that a parent topic sentence is more similar to its children than the sentences descending from the other parents. To confirm this property, we estimated the similarity of sentences in parent-child pairs and non parent-child pairs. To measure sentence similarity, we used *ALBERT* (Lan et al., 2019), which is a SoTA model on the semantic textual similarity benchmark (**STS-B**; Cer et al., 2017). In our experiment, we used ALBERT-base, which achieves a 84.7 Pearson correlation coefficient against the test sets of STS-B. As shown in Table 7, parent-child sentence pairs are more similar than those of non parent-child pairs in both datasets. This result indicates that the generated sentences linked by parent-child relations are topically coherent.

Hierarchical Affinity	Yelp	Amazon
Parent-child pairs	2.39	1.33
Non parent-child pairs	1.59	0.76

Table 7: Average sentence similarity of the topic sentence pairs, ranging from 0 (different) to 5 (similar).

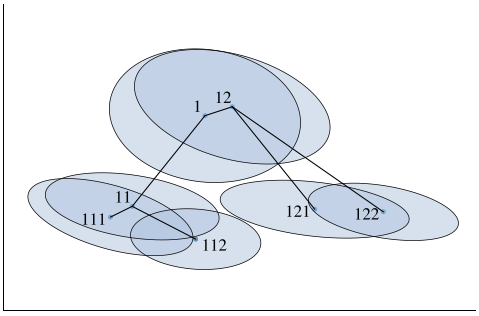
Topic Specialization	Yelp	Amazon
First level	1.68	1.60
Second level	1.99	1.63
Third level	2.16	1.84

Table 8: Average specialization score of each level topics, ranging from 1 (general) to 5 (specific).

Topic Specialization. In tree-structured topics, we would expect the root topic to generate general sentences, whereas more specific content is conveyed by the sentences generated by the leaf topics. To empirically test this property, we estimated the average specificity of sentences at each level of the tree-structured topics. We fine-tuned ALBERT-base on the task of estimating the specificity of sentences (Louis and Nenkova, 2011). We used the dataset provided by Ko et al. (2019), which comprises the Yelp, Movie, and Tweet domains. The fine-tuned model achieves a SoTA performance of 86.2 Pearson correlation coefficient on the test sets in Yelp. As shown in Table 8, we see that sentences with lower topics are more specific than higher topics. This indicates that the root sentences refer to general topics, whereas leaf sentences describe more specific topics.

5.4 Analyzing Latent Space of Sentences

In Figure 5, we project the latent code of topic sentences of a restaurant review onto the top two principal component vector space. Following the modeling assumption, the latent distributions of child sentences are located relatively near their parent distributions. This property ensures that the parent and child sentences are topically coherent, as shown in Table 7. Furthermore, we present



1. A great thing to go to the place.
   - 11. Friendly staff and I appreciated the food.
     - 111. The staff is very friendly and knowledgable.
     - 112. Food was great, and the service was great.
   - 12. There is a lot of a great meal.
     - 121. The meal was great, but the food was fantastic.
     - 122. The <unk> was delicious and the fish tacos were great.

Figure 5: 2-D latent space projected by principal component analysis. Each point corresponds to the mean of the latent distribution of a topic sentence, and each circle denotes the same Mahalanobis distance from the mean.

LogDetCov	Yelp	Amazon
First level	28.83	27.21
Second level	26.22	26.54
Third level	23.28	24.55

Table 9: Average log determinant of covariance matrices (LogDetCov) on each level.

the average log determinant of the covariance matrices at each level in Table 9. We confirm that the latent code of the topic sentences has a smaller variance towards the leaves. This property forces the topic sentences to be more specific as the level becomes deeper, as described in Table 8.

6 Related Work

6.1 Text Generation with Topic Guidance

The VAE is intensively used to obtain disentangled latent code of sentences (Bowman et al., 2016; Hu et al., 2017; Tang et al., 2019). Closely related to our work, Wang et al. (2019) specify the prior as a GMM, where each mixture component corresponds to the latent code of a topic sentence and is mixed with the topic distribution inferred by the flat neural topic model (Miao et al., 2017).

In contrast, we address a novel challenge to generate topic sentences with tree-structured topic guidance, where the root sentence refers to a general topic, whereas the leaf sentences describe more specific topics. We adopt the tree-structured neural topic model (Isonuma et al., 2020) to infer the topic distribution of sentences and introduce a recursive Gaussian mixture prior for modeling the latent distribution of sentences in a document.

6.2 Unsupervised Summary Generation

Owing to the success of supervised abstractive summarization by neural architectures (Nallapati et al., 2016; See et al., 2017; Liu and Lapata, 2019a), unsupervised sentence compression (Fevry and Phang, 2018; Baziotis et al., 2019), and unsupervised summary generation (Isonuma et al., 2019) have recently drawn attention.

Recently, specifically for opinionated texts, several abstractive multi-document summarization methods have been developed, such as MeanSum, Copycat, and DenoiseSum, as explained in Section 4.3. Concurrently with our work, Angelidis et al. (2021) use quantitized transformers enabling aspect-based extractive summarization, and Amplayo et al. (2020) incorporate the aspect and sentiment distributions into the unsupervised abstractive summarization. Our method incorporates topic-tree structure into unsupervised abstractive summarization and generates summaries consisting of multiple granularities of topics.

7 Conclusion

In this paper, we proposed a novel unsupervised abstractive opinion summarization method by generating topic sentences with tree-structured topic guidance. Experimental results demonstrated that the generated summaries are more informative and cover more input content than those generated by the recent unsupervised summarization (Bražinskas et al., 2020). Additionally, we demonstrated that the variance of latent Gaussians represents the granularity of sentences, analogous to Gaussian word embedding (Vilnis and McCallum, 2015). This property will be useful not only for summarization but also for other

tasks that need to consider the granularity of the contents.

Acknowledgments

We would like to thank the anonymous reviewers and action editor, Asli Celikyilmaz, for their valuable feedback. This work was supported by JST ACT-X grant number JPMJAX1904 and JSPS KAKENHI grant number JP20J10726, Japan.

References

David Alvarez-Melis and Tommi S. Jaakkola. 2017. Tree-structured decoding with doubly-recurrent neural networks. In *Proceedings of the 5th International Conference on Learning Representations*.

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2020. Unsupervised opinion summarization with content planning. *CoRR*, arXiv:2012.07808v1. https://doi.org/10.18653/v1/2020.acl-main.175

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.

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. https://doi.org/10.1162/tacl_a_00366

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

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34. https://doi.org/10.1162/coli.2008.34.1.1

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. Seq^3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 673–681.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21. https://doi.org/10.18653/v1/K16-1002

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. https://doi.org/10.18653/v1/2020.acl-main.461

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336. https://doi.org/10.1145/290941.291025

Giuseppe Carenini, Jackie Chi Kit Cheung, and Adam Pauls. 2013. Multi-document summarization of evaluative text. *Computational Intelligence*, 29(4):545–576. https://doi.org/10.1111/j.1467-8640.2012.00417.x

Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 815–824.

Asli Celikyilmaz and Dilek Hakkani-Tur. 2011. Discovery of topically coherent sentences for extractive summarization. In *Proceedings of the 49th Annual Meeting of the Association for*

Computational Linguistics: Human Language Technologies, pages 491–499.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 1–14.

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*, pages 1223–1232.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of NIPS 2014 Workshop on Deep Learning*.

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*, pages 4171–4186.

Günes Erkan and Dragomir R. Radev. 2004. Lexpagerank: Prestige in multi-document text summarization. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 365–371.

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

Thibault Fevry and Jason Phang. 2018. Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422. https://doi.org/10.18653/v1/K18-1040

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*, pages 340–348.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1602–1613. https://doi.org/10.3115/v1/D14-1168

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18. https://doi.org/10.3115/1611638.1611640

Thomas L. Griffiths, Michael I. Jordan, Joshua B. Tenenbaum, and David M. Blei. 2004. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*, pages 17–24.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1587–1596.

Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. 2017. Extractive summarization using multi-task learning with document classification. In *Proceedings of the 2017 Conference on empirical methods in natural language processing*, pages 2101–2110. https://doi.org/10.18653/v1/D17-1223

Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2020. Tree-structured neural topic model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 800–806.

https://doi.org/10.18653/v1/2020.acl-main.73

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

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of the 5th International Conference on Learning Representations*.

Joon Hee Kim, Dongwoo Kim, Suin Kim, and Alice Oh. 2012. Modeling topic hierarchies with the recursive chinese restaurant process. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 783–792.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, arXiv:1412.6980v9.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations*.

Svetlana Kiritchenko and Saif M. Mohammad. 2016. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 811–817. https://doi.org/10.18653/v1/N16-1095

Wei-Jen Ko, Greg Durrett, and Junyi Jessy Li. 2019. Domain agnostic real-valued specificity prediction. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6610–6617. https://doi.org/10.1609/aaai.v33i01.33016610

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the 7th International Conference on Learning Representations*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081. https://doi.org/10.18653/v1/P19-1500

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3721–3731. https://doi.org/10.18653/v1/D19-1387

Annie Louis and Ani Nenkova. 2011. Automatic identification of general and specific sentences by leveraging discourse annotations. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 605–613.

Jordan J. Louviere, Terry N. Flynn, and Anthony Alfred John Marley. 2015. *Best-Worst Scaling: Theory, Methods and Applications*, Cambridge University Press. https://doi.org/10.1017/CBO9781107337855

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. https://doi.org/10.18653/v1/D15-1166

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the 5th International Conference on Learning Representations*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919. https://doi.org/10.18653/v1/2020.acl-main.173

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. https://doi.org/10.1145/2766462.2767755

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2410–2419.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. https://doi.org/10.18653/v1/K16-1028

John Paisley, Chong Wang, David M. Blei, and Michael I. Jordan. 2014. Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270. https://doi.org/10.1109/TPAMI.2014.2318728

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1278–1286.

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*, pages 1073–1083.

Hongyin Tang, Miao Li, and Beihong Jin. 2019. A topic augmented text generation model: Joint learning of semantics and structural features. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5093–5102. https://doi.org/10.18653/v1/D19-1513

Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 308–316.

Luke Vilnis and Andrew McCallum. 2015. Word representations via Gaussian embedding. In *Proceedings of the 3rd International Conference on Learning Representations*.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational auto-encoder for text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 166–177. https://doi.org/10.18653/v1/N19-1015

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3881–3890.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1097–1100.

A Appendices

A.1 Inference of Topic Distribution

To approximate the tree-structured topic distribution of a sentence, we use a tree-structured neural

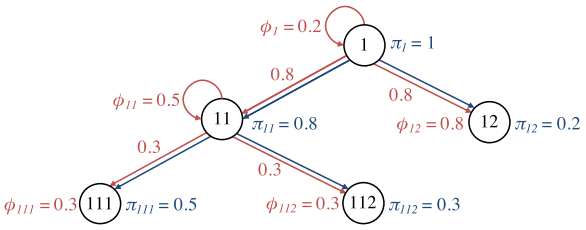


Figure 6: Example of a path distribution (blue) and level distribution (red). Both the sum of a path distribution over each level and the sum of a level distribution over each path are equal to 1.

topic model (**TSNTM**; Isonuma et al., 2020), which transforms a sentence into a tree-structured topic distribution using neural networks. While their model is based on the nested Chinese restaurant process (**nCRP**; Griffiths et al., 2004), we make a minor change to use the nested hierarchical Dirichlet process (**nHDP**; Paisley et al., 2014). The nHDP generates a sentence-specific path distribution $\boldsymbol{\pi}_s$ and level distribution $\boldsymbol{\phi}_s$ as

$$\nu_{s,k} \sim \text{Beta}(1,\gamma), \pi_{s,k} = \pi_{s,par(k)} \nu_{s,k} \prod_{j \in \text{Sib}(k)} (1-\nu_{s,j}) \tag{20}$$

$$\eta_{s,k} \sim \text{Beta}(\alpha,\beta), \phi_{s,k} = \eta_{s,k} \prod_{j \in \text{Anc}(k)} (1-\eta_{s,j}) \tag{21}$$

$$\theta_{s,k} = \pi_{s,k} \cdot \phi_{s,k} \tag{22}$$

where $\text{Sib}(k)$ and $\text{Anc}(k)$ are the sets of the k-th topic's preceding-siblings and ancestors, respectively. As described in Figure 6, $\pi_{s,k}$ denotes the probability that a sentence s selects a path from the root to the k-th topic. $\phi_{s,k}$ denotes the probability that a sentence s does not select the ancestral topics $j \in \text{Anc}(k)$ but remains in the k-th topic along the path. By multiplying these two probabilities, we obtain $\theta_{s,k}$; the probability that a sentence s selects the topic k. The nHDP does not make a significant difference in the summarization performance from the nCRP. However, the nHDP permits different lengths of each path, whereas the nCRP restricts each path length to be the same.

Following Isonuma et al. (2020), we use the doubly recurrent neural networks (**DRNN**; Alvarez-Melis and Jaakkola, 2017) to transform a sentence embedding $\boldsymbol{y}_s = \text{RNN}(\boldsymbol{w}_s)$ to the path distribution $\boldsymbol{\pi}_s$ and level distribution $\boldsymbol{\phi}_s$. The DRNN consists of two RNN decoders over respectively the ancestors and siblings. We compute the k-th topic's hidden state $\boldsymbol{h}_k$ using (23) and obtain the path distribution by alternating $\boldsymbol{\nu}_s$ as (24):

$$\boldsymbol{h}_k = \tanh(\boldsymbol{W}_p \boldsymbol{h}_{par(k)} + \boldsymbol{W}_s \boldsymbol{h}_{k-1}) \tag{23}$$

$$\nu_{s,k} = \text{sigmoid}(\boldsymbol{h}_k^\top \boldsymbol{y}_s) \tag{24}$$

where $\boldsymbol{h}_{par(k)}$ and $\boldsymbol{h}_{k-1}$ are the hidden states of a parent and a previous sibling of the k-th topic, respectively. Similarly, we obtain the level distribution, $\boldsymbol{\phi}_s$, by computing $\boldsymbol{\eta}_s$ with another DRNN.

# of topics for each level (total)	R-1	R-2	R-L
1–2–4 (7)	29.03	4.39	16.94
1–3–9 (13)	31.42	4.43	17.19
1–4–16 (21)	33.24	5.15	18.01
1–5–25 (31)	31.94	4.78	17.50
1–6–36 (43)	33.25	4.82	17.81

Table 10: Sensitivity for various number of branches.

# of topics for each level (total)	R-1	R-2	R-L
1–3 (4)	23.63	2.38	14.35
1–3–9 (13)	31.42	4.43	17.19
1–3–9–27 (40)	32.55	4.75	17.70

Table 11: Sensitivity for various number of levels.

A.2 Sensitivity for the Number of Topics

We investigated how the number of topics affects summarization performance. Table 10 shows the ROUGE scores on the various number of branches with a fixed depth of 3 in topic-tree structure. When the number of topics is small, the models achieve a relatively low score. However, when the number of branches ≥ 4, the performance does not significantly change for various numbers of topics. A similar trend is confirmed in Table 10, which shows the ROUGE scores on the various number of levels with the fixed number of branches of 3. These results indicate that our model is relatively robust for the number of topics.

A.3 Derivation of Equation (19)

Proposition: when $q(\boldsymbol{x}_s|z_s)$ is given by (12), (25) holds:

$$\sum\nolimits_s \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\big[q(\boldsymbol{x}_s|\boldsymbol{w}_s)|p(\boldsymbol{x}_s|z_s=k)\big] - \sum\nolimits_s \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\big[q(\boldsymbol{x}_s|z_s=k)|p(\boldsymbol{x}_s|z_s=k)\big] \geq 0 \tag{25}$$

Proof: The first term of (25) is re-written as:

$$\begin{aligned}
&\sum\nolimits_s \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\big[q(\boldsymbol{x}_s|\boldsymbol{w}_s)|p(\boldsymbol{x}_s|z_s=k)\big] \\
=&\sum\nolimits_s \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\big[\mathcal{N}(\hat{\boldsymbol{\mu}}_s, \hat{\boldsymbol{\Sigma}}_s)|\mathcal{N}(\hat{\boldsymbol{\mu}}_{d,par(k)}, \hat{\boldsymbol{\Sigma}}_{d,par(k)})\big] \\
=&\frac{1}{2}\sum\nolimits_s \hat{\theta}_{s,k}\Big\{\log|\hat{\boldsymbol{\Sigma}}_{d,par(k)}| - \log|\hat{\boldsymbol{\Sigma}}_s| + \mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\Sigma}}_s] \\
&+(\hat{\boldsymbol{\mu}}_s - \hat{\boldsymbol{\mu}}_{d,par(k)})^\top \hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}(\hat{\boldsymbol{\mu}}_s - \hat{\boldsymbol{\mu}}_{d,par(k)}) - n\Big\} \\
=&\frac{1}{2}\sum\nolimits_s \hat{\theta}_{s,k}\Big\{C_{d,par(k)} - \log|\hat{\boldsymbol{\Sigma}}_s| + \mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\Sigma}}_s] \\
&+\hat{\boldsymbol{\mu}}_s^\top \hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\mu}}_s - 2\hat{\boldsymbol{\mu}}_{d,par(k)}^\top \hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\mu}}_s\Big\} \\
=&\frac{1}{2}\sum\nolimits_s \hat{\theta}_{s,k}\Big\{C_{d,par(k)} - \log|\hat{\boldsymbol{\Sigma}}_s| + \mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\Sigma}}_s] \\
&+\mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\mu}}_s\hat{\boldsymbol{\mu}}_s^\top] - 2\hat{\boldsymbol{\mu}}_{d,par(k)}^\top \hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\mu}}_{d,k}\Big\}
\end{aligned} \tag{26}$$

as $\sum_s \hat{\theta}_{s,k}\hat{\boldsymbol{\mu}}_{d,k} = \sum_s \hat{\theta}_{s,k}\hat{\boldsymbol{\mu}}_s$ from (13).

The second term is similarly expanded as:

$$\begin{aligned}
&\sum\nolimits_s \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\big[q(\boldsymbol{x}_s|z_s=k)|p(\boldsymbol{x}_s|z_s=k)\big] \\
=&\frac{1}{2}\sum\nolimits_s \hat{\theta}_{s,k}\Big\{C_{d,par(k)} - \log|\hat{\boldsymbol{\Sigma}}_{d,k}| + \mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\Sigma}}_{d,k}] \\
&+\mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\mu}}_{d,k}\hat{\boldsymbol{\mu}}_{d,k}^\top] - 2\hat{\boldsymbol{\mu}}_{d,par(k)}^\top \hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}\hat{\boldsymbol{\mu}}_{d,k}\Big\}
\end{aligned} \tag{27}$$

Therefore, (25) is arranged as:

$$\begin{aligned}
&\textstyle\sum_s \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\big[q(\boldsymbol{x}_s|\boldsymbol{w}_s)|p(\boldsymbol{x}_s|z_s=k)\big] \\
&\textstyle-\sum_s \hat{\theta}_{s,k} \mathrm{D}_{\mathrm{KL}}\big[q(\boldsymbol{x}_s|z_s=k)|p(\boldsymbol{x}_s|z_s=k)\big] \\
=&\frac{1}{2}\sum\nolimits_s \hat{\theta}_{s,k}\Big\{-\log|\hat{\boldsymbol{\Sigma}}_s| + \log|\hat{\boldsymbol{\Sigma}}_{d,k}| \\
&+\mathrm{Tr}\big[\hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1}(\hat{\boldsymbol{\Sigma}}_s + \hat{\boldsymbol{\mu}}_s\hat{\boldsymbol{\mu}}_s^\top - \hat{\boldsymbol{\Sigma}}_{d,k} - \hat{\boldsymbol{\mu}}_{d,k}\hat{\boldsymbol{\mu}}_{d,k}^\top)\big]\Big\} \\
=&\frac{1}{2}\sum\nolimits_s \hat{\theta}_{s,k}\Big\{-\log|\hat{\boldsymbol{\Sigma}}_s| + \log|\hat{\boldsymbol{\Sigma}}_{d,k}|\Big\} + \frac{1}{2}\Big\{\mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{d,par(k)}^{-1} \\
&\textstyle\sum_s \hat{\theta}_{s,k}(\hat{\boldsymbol{\Sigma}}_s + \hat{\boldsymbol{\mu}}_s\hat{\boldsymbol{\mu}}_s^\top - \hat{\boldsymbol{\Sigma}}_{d,k} - \hat{\boldsymbol{\mu}}_{d,k}\hat{\boldsymbol{\mu}}_{d,k}^\top)]\Big\} \\
=&\frac{1}{2}\sum\nolimits_s \hat{\theta}_{s,k}\Big\{-\log|\hat{\boldsymbol{\Sigma}}_s| + \log|\hat{\boldsymbol{\Sigma}}_{d,k}|\Big\}
\end{aligned} \tag{28}$$

as $\sum_s \hat{\theta}_{s,k}\big\{\hat{\boldsymbol{\Sigma}}_s + \hat{\boldsymbol{\mu}}_s\hat{\boldsymbol{\mu}}_s^\top\big\} = \sum_s \hat{\theta}_{s,k}\big\{\hat{\boldsymbol{\Sigma}}_{d,k} + \hat{\boldsymbol{\mu}}_{d,k}\hat{\boldsymbol{\mu}}_{d,k}^\top\big\}$ from (14). The given equation eventually comes down to a comparison of the entropy.

Since, in general, $-\int q_1(\boldsymbol{x}) \log q_2(\boldsymbol{x})\ d\boldsymbol{x} \geq -\int q_1(\boldsymbol{x}) \log q_1(\boldsymbol{x})\ d\boldsymbol{x}$ holds, we obtain (29):

$$\begin{aligned}
&\textstyle\sum_s \hat{\theta}_{s,k}\Big\{-\int q(\boldsymbol{x}_s|\boldsymbol{w}_s) \log q(\boldsymbol{x}_s|z_s=k)\ d\boldsymbol{x}_{\boldsymbol{s}}\Big\} \\
\geq&\textstyle\sum_s \hat{\theta}_{s,k}\Big\{-\int q(\boldsymbol{x}_s|\boldsymbol{w}_s) \log q(\boldsymbol{x}_s|\boldsymbol{w}_s)\ d\boldsymbol{x}_{\boldsymbol{s}}\Big\}
\end{aligned} \tag{29}$$

As the right term is a weighted sum of the normal distribution entropy, it can be rewritten as:

$$\begin{aligned}
&\textstyle\sum_s \hat{\theta}_{s,k}\Big\{-\int q(\boldsymbol{x}_s|\boldsymbol{w}_s) \log q(\boldsymbol{x}_s|\boldsymbol{w}_s)\ d\boldsymbol{x}_{\boldsymbol{s}}\Big\} \\
=&\frac{1}{2}\sum\nolimits_s \hat{\theta}_{s,k}\Big\{\log|\hat{\boldsymbol{\Sigma}}_s| + n\log 2\pi + n\Big\}
\end{aligned} \tag{30}$$

Meanwhile, we can expand the left term as:

$$\begin{aligned}
&\textstyle\sum_s \hat{\theta}_{s,k}\Big\{-\int q(\boldsymbol{x}_s|z_s=k) \log q(\boldsymbol{x}_s|\boldsymbol{w}_s)\ d\boldsymbol{x}_{\boldsymbol{s}}\Big\} \\
=&\frac{1}{2}\sum\nolimits_s \hat{\theta}_{s,k}\Big\{\log|\hat{\boldsymbol{\Sigma}}_{d,k}| + n\log 2\pi \\
&+\mathrm{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)}[(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})^\top \hat{\boldsymbol{\Sigma}}_{d,k}^{-1}(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})]\Big\}
\end{aligned} \tag{31}$$

The last term in (31) is expressed as:

$$\begin{aligned}
&\textstyle\sum_s \hat{\theta}_{s,k}\Big\{\mathrm{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)}\big[(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})^\top \hat{\boldsymbol{\Sigma}}_{d,k}^{-1}(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})\big]\Big\} \\
=&\textstyle\sum_s \hat{\theta}_{s,k}\Big\{\mathrm{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)}[\mathrm{Tr}(\hat{\boldsymbol{\Sigma}}_{d,k}^{-1}(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})^\top)]\Big\} \\
=&\mathrm{Tr}\Big[\hat{\boldsymbol{\Sigma}}_{d,k}^{-1}\textstyle\sum_s \hat{\theta}_{s,k}\{\mathrm{E}_{q(\boldsymbol{x}_s|\boldsymbol{w}_s)}[(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}_{d,k})^\top]\}\Big] \\
=&\mathrm{Tr}\Big[\hat{\boldsymbol{\Sigma}}_{d,k}^{-1}\textstyle\sum_s \hat{\theta}_{s,k}\hat{\boldsymbol{\Sigma}}_{d,k}\Big] \\
=&\textstyle\sum_s \hat{\theta}_{s,k}\, n
\end{aligned} \tag{32}$$

Thus, by combining (29), (30), (31), (32), $\sum_s \hat{\theta}_{s,k}\big\{\log|\hat{\boldsymbol{\Sigma}}_{d,k}|\big\} \geq \sum_s \hat{\theta}_{s,k}\big\{\log|\hat{\boldsymbol{\Sigma}}_s|\big\}$ holds and implies (25).