HYDRASUM: Disentangling Style Features in Text Summarization with Multi-Decoder Models

Tanya Goyal[1] **Nazneen Rajani**[2] **Wenhao Liu**[3] **Wojciech Kryściński**[4]
[1] Department of Computer Science, The University of Texas at Austin
[2] Hugging Face [3] Faire [4] Salesforce Research
tanyagoyal@utexas.edu

Abstract

Summarization systems make numerous "decisions" about summary properties during inference, e.g. degree of copying, specificity and length of outputs, etc. However, these are implicitly encoded within model parameters and specific styles cannot be enforced. To address this, we introduce HYDRASUM, a new summarization architecture that extends the single decoder framework of current models to a mixture-of-experts version with multiple decoders. We show that HYDRASUM's multiple decoders automatically learn contrasting summary styles when trained under the standard training objective without any extra supervision. Through experiments on three summarization datasets (CNN, NEWSROOM and XSUM), we show that HYDRASUM provides a simple mechanism to obtain stylistically-diverse summaries by sampling from either individual decoders or their mixtures, outperforming baseline models. Finally, we demonstrate that a small modification to the gating strategy during training can enforce an even stricter style partitioning, e.g. high- vs low-abstractiveness or high- vs low-specificity, allowing users to sample from a larger area in the generation space and vary summary styles along multiple dimensions.[1]

1 Introduction

Abstractive summarization (Rush et al., 2015; See et al., 2017) involves a combination of generation decisions, such as what content to directly copy from the input and what content to paraphrase, the level of specificity vs generality, length, readability, etc. of generated summaries. Current summarization systems (Lewis et al., 2020; Zhang et al., 2020) implicitly encode these decisions in their parameters, but provide no mechanism for end users to specify their stylistic preferences. Commonly used decoding methods such as beam search, top-k decoding (Fan et al., 2018b) or diverse decoding (Vijayakumar et al., 2018) tend to generate stylistically similar outputs, and cannot be queried for multiple diverse summaries without sacrificing quality. Prior work in style transfer (Hu et al., 2017; Krishna et al., 2020) target styles that are not relevant to summarization (e.g. sentiment, Shakespearean language, etc.) and use explicit interventions to enforce style. Instead, we ask: **what style combinations naturally occur in abstractive summarization datasets and can models automatically disentangle them?**

In this paper, we propose HYDRASUM - a new summarization architecture that disentangles the different stylistic decisions made by abstractive summarization models from the models weights into an explicit model component. Our model contains a single transformer-based encoder to encode the input document and a mixture-of-experts with multiple decoders for summary generation. At each time step of the generation phase, the next token's probability distribution is computed by combining the output probabilities of all individual decoders. This allows our model to distribute the diverse stylistic and lexical features encountered in the training data, even those within the same reference summary, across the parameters of separate decoders. During inference, we leverage the modularity in the decoder framework to sample from these individual decoders, each of which generates stylistically-distinct summaries.

As a toy example, consider a 2-decoder scenario in which one decoder learns to only copy phrases or words from the input document, while the second decoder only learns paraphrasing and syntactic transformations. While individual decoders cannot cover the range of stylistic variations in the dataset, a weighted combination or mixture of the two decoders can be used to model the summarization dataset. In practice, we found that this partitioning of summarization "skills" between decoders

[1] Code and model checkpoints are shared at https://github.com/salesforce/hydra-sum.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 464–479
December 7-11, 2022

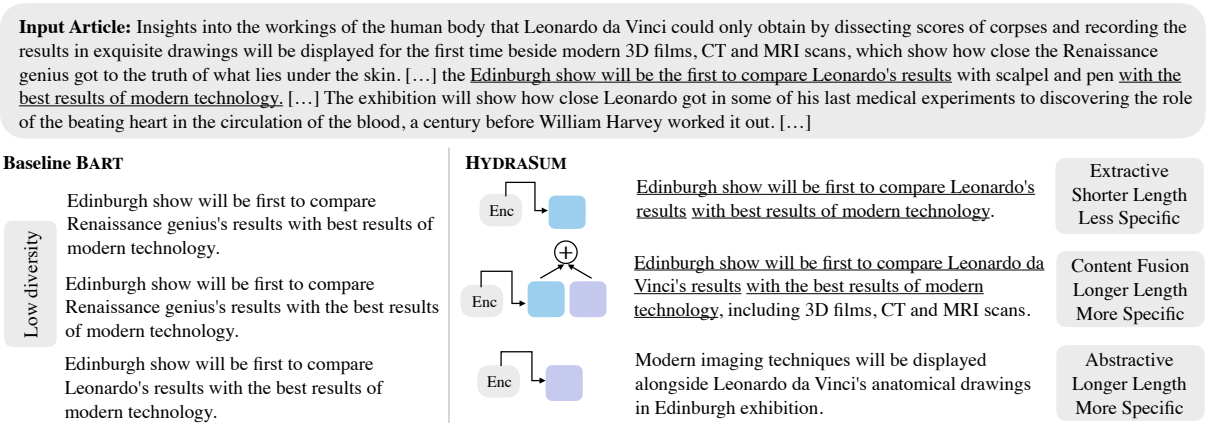


Figure 1: Examples of generated summaries for a NEWSROOM article using both BART and a 2-decoder HYDRASUM model. Longer copied sequences (denoting extractive behavior) are underlined. For HYDRASUM, summaries from different mixtures of decoders differ in degree of abstractiveness, specificity, and length.

is much fuzzier, and occurs along multiple dimensions such as degree of abstractiveness (copying), readability, specificity and length. Figure 1 shows examples of summaries generated by the baseline model and a 2-decoder version of HYDRASUM. We see that HYDRASUM produces a more stylistically distinct set of summaries by varying the degree of abstractiveness and summary length, or including additional details such as *3D films, CT and MRI scans* to vary specificity. On the other hand, baseline BART exhibits low diversity and largely generates extractive summaries (See et al., 2017; Goyal and Durrett, 2021).

Our contributions in this paper are: (1) We show that our proposed HYDRASUM model automatically assigns distinct summary "*skills*" to different decoders during training, for both 2- and 3-decoder versions across three summarization datasets (Section 3.1). (2) We show that this property can be operationalized to obtain multiple summaries exhibiting better stylistic diversity and Top-K quality compared to baseline models (Section 3.3). (3) Finally, we demonstrate that a simple data preprocessing and gating strategy during training can be used to explicitly dictate *which* feature is partitioned across different decoders. Not only does this allow us to enforce a greater style difference between decoders compared to prompt-based baselines, it also provides a mechanism for multi-style variation in summary generation (Section 4).

2 Methodology

Current state-of-the-art summarization models (e.g. BART, PEGASUS) use transformer-based encoder-

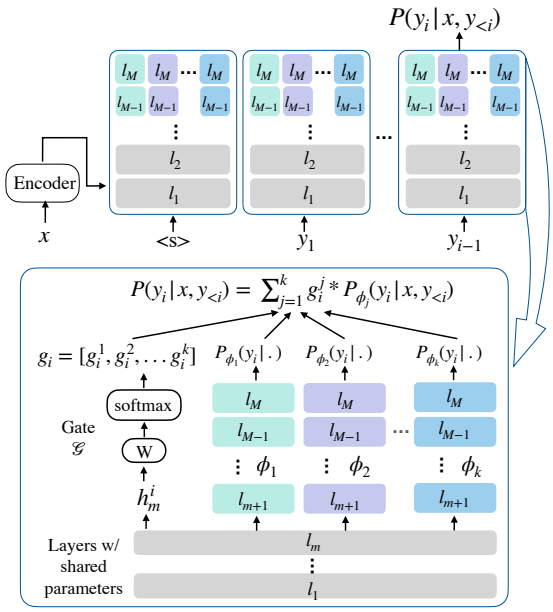


Figure 2: Our proposed HYDRASUM architecture. The decoder network of standard models is modified to incorporate multiple decoders. The lower layers of these decoders have shared parameters and a gating mechanism is used to combine their output probabilities in a mixture-of-experts formulation.

decoder architectures. Similarly to those models, HYDRASUM consists of an encoder network that accepts the document x as input. The decoder network, however, is modified to incorporate $k(> 1)$ decoders, $\phi_1, \phi_2, ...\phi_k$, as depicted in Figure 2. At time step i, each decoder outputs a probability distribution $P_{\phi_k}(y_i|x, y_{<i})$ over the vocabulary, corresponding to the next-token probabilities. The final output probability $P(y_i|x, y_{<i})$ is computed as a mixture of these k probability distributions,

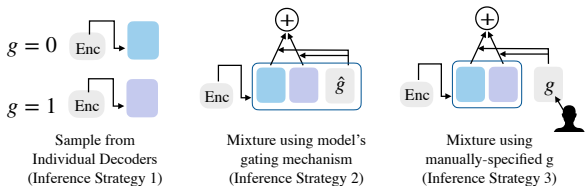


Figure 3: HYDRASUM's inference options.

with the mixing coefficients predicted by a gating mechanism $\mathcal{G}$.

Multi-Decoder Architecture Let M be the total number of decoder blocks in a single decoder: e.g. $M = 12$ for BART-LARGE. In HYDRASUM, the parameters of the $m(< M)$ bottom layers are shared between the k decoders. This reduces the number of extra parameters introduced into the model architecture. The top $M - m$ layers of the different decoders are independently trained. The right block of Figure 2 shows a detailed view of the multi-decoder architecture at a single time step i.

Gating Mechanism A gating mechanism $\mathcal{G}$ is used to combine the output distributions of the k decoders. Let h_i^m be the hidden state output of the m^{th} decoder layer at time step i, i.e. the output of the last shared layer. We use this hidden state representation to obtain the coefficients for our mixture of experts. The representation h_i^m is fed into a feed forward layer W (size $= (|h_i^m|, k)$), followed by a softmax layer. This outputs a probability distribution g_i which is used to compute the overall next-token output probability as follows: $P(y_i|x, y_{<i}) = \sum_{j=1:k} g_i^j * P_{\phi_j}(y_i|x, y_{<i})$. Here, g_i^j is the probability of selecting the j^{th} decoder at time step i.

Training Similar to standard summarization models, the HYDRASUM architecture is trained to minimize the cross entropy loss of the reference summaries, conditioned on the input document: $loss = -\sum_i \log P(y_i|x, y_{<i})$. The model implicitly decides the contribution of each decoder to the final output probability, i.e. g_i^j for decoder j at time step i, using the gating mechanism $\mathcal{G}$ from above.

2.1 Inference

HYDRASUM provides several options of output distributions which differ in how the mixture weights are obtained (see Figure 3). During inference, we can sample from these different options, or **inference strategies**, to generate summaries:

1. **Individual Decoders**: To generate summaries using only the j^{th} decoder, the output of the gating mechanism is overridden with $[0, 0..., 1, ..., 0]$ where $g^j = 1$ and $g^{i \neq j} = 0$ for all time steps.

2. **Mixture using $\mathcal{G}$**: The mixture weights are decided by the model, i.e. $g_i^j = (W^T h_i^m)_j$ for decoder ϕ_j at time step i.

3. **Mixture with manually-specified g**: Consider a 2-decoder HYDRASUM model, where decoder 0 learns abstractive and decoder 1 learns extractive features. The degree of abstraction can be varied by specifying the contribution of individual decoders through gate coefficients $[1 - g, g]$. Effectively, this modifies the output probability to: $P(y_i|\cdot) = (1-g) * P_{\phi_0}(y_i|\cdot) + g * P_{\phi_1}(y_i|\cdot)$.

3 Experiments

We conduct experiments on three news summarization datasets: CNN (Hermann et al., 2015; Nallapati et al., 2016), NEWSROOM[2] (Grusky et al., 2018) and XSUM (Narayan et al., 2018). The reference summaries in these datasets exhibit a mutually-distinct stylistic properties and help evaluate HYDRASUM's capabilities under these distinct test conditions.

For all experiments, BART-LARGE (Lewis et al., 2020) is used as the model initialization: in a k-decoder variant of HYDRASUM, all k decoders are initialized with the weights of BART-LARGE's decoder. The weights of the gating mechanism $\mathcal{G}$ are randomly initialized from a normal distribution $\mathcal{N}(0, 0.02)$. We set the number of shared layers, i.e. m to 8, for all experiments.[3] Our model architecture is implemented using the Huggingface Library (Wolf et al., 2020). More training and inference details are in Appendix A.

We compare against the standard BART-based summarization baseline. For XSUM, we use the publicly available BART-LARGE-XSUM checkpoint. For CNN and NEWSROOM, we fine-tune the BART-LARGE checkpoint on their corresponding training datasets ourselves.[4] Beam decoding is used to generate summaries for all models.

[2]We run experiments on the *mixed* subset of NEWSROOM to limit data size. We found that this subset was less noisy and more diverse than the *abstractive* and *extractive* subsets.

[3]Experiments with other values of $m(= 6, 10)$ are in Appendix B. Varying m does not alter our conclusions.

[4]Publicly available BART-LARGE-CNN (Lewis et al., 2020) and PEGASUS-NEWSROOM (Zhang et al., 2020) trained on the full CNNDM and NEWSROOM datasets perform poorly

3.1 Style Partitioning

First, we investigate whether individual HYDRASUM decoders learn different styles when trained using the standard training objective? If yes, which stylistic features vary across different decoders?

Metrics We measure *style* along the following summarization-relevant dimensions:

1. **Abstractiveness**: We follow Grusky et al. (2018) and report two metrics, *coverage* which denotes the fraction of summary words that are also present in the input, and *density* which denotes the average length of copied contiguous spans in a summary. Additionally, we report the 2-gram overlap between the generated summary and the input article.

2. **Degree of specificity** of generated summaries, quantified using the Speciteller tool (Li and Nenkova, 2015). To align with their definition, we segment summaries into sentences and report the macro-average of the sentence-level specificity across all summaries.

3. **Length metrics**: We report two metrics for this, *absolute length* (number of words) of generated summaries, and *compression ratio*, computed as the ratio of the number of words in the summary and the input article.

4. **Readability** scores of generated summaries, measured using the Flesch readability ease test (Flesch, 1948).

In addition to these style-based metrics, we report **Quality**, measured by ROUGE (Lin, 2004) scores of the generated summaries with respect to the reference summaries.

For analysis, we generate 3 summaries for each input: using individual decoders D0 and D1 (Inference Strategy 1, see Section 2.1), and the mixture model (Mix) where the mixture weights are obtained using the gating mechanism $\mathcal{G}$ (Strategy 2). The latter corresponds to sampling from the HYDRASUM's actual output distribution.

3.2 Results

Style differences between decoders Differences in style between D0 and D1 are outlined in Table 1. Features for which this difference is significant, i.e. $p < 0.05$ according to the bootstrap re-sampling test, are highlighted in gray. For both CNN and NEWSROOM, significant differences are observed along the abstractiveness and specificity metrics. Moreover, summaries for CNN also differ along other metrics such as length and readability. The least amount of style difference is observed for XSUM where the decoders only differ in specificity, although this difference (approx. .13) is more than the other datasets. We hypothesize that the similarity in abstractiveness levels of the XSUM decoders is due to the low diversity along this feature in XSUM's training data. These results indicate that although HYDRASUM's training encourages the two decoders to learn distinct styles, the combination of features along which they differ is heavily dependent on the datasets themselves.

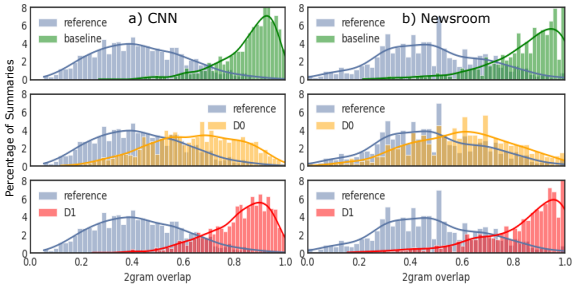


Figure 4: Graphs plot the 2gram overlap of the baseline and HYDRASUM decoders. Compared to the baseline, D0 decoder samples summaries from a distribution that more closely resembles the reference distribution.

Coverage over the generation space Interestingly, for both CNN and NEWSROOM, we observe that the baseline model fails to cover the entire range of abstractive behavior seen in the reference summaries. Figure 4 demonstrates this; the top graphs plot the 2-gram overlap of the reference summaries and the baseline BART summaries, showing substantial mismatch. The references are more diverse, while BART summaries are highly extractive. This is a known issue with standard training (See et al., 2017; Goyal et al., 2022); summarization models tend to overfit on the easier extractive examples and do not learn from the abstractive examples. HYDRASUM addresses this limitation by encouraging the two decoders to learn contrasting levels of abstractiveness. Figure 4 shows that the D0 decoders for both datasets generate abstractive summaries that more closely resembles the reference distribution. Meanwhile, D1 generates extractive summaries, collectively providing better coverage over the abstractiveness space. Later,

on the CNN only and NEWSROOM-MIXED only test sets used in our work. Hence, we re-train these.

		Abstractiveness			Specificity	Length-metrics		Readability	Quality
		Coverage	Density	2G Overlap		Abs.	Comp.	FRE	R1/R2/RL
CNN	Ref	0.85	3.14	0.43	0.44	37.33	0.07	52.51	-
	Baseline	0.97	10.33	0.80	0.44	50.71	0.10	54.03	34.87/**14.88**/31.82
	D0	0.93	5.69	0.64	0.48	46.07	0.09	58.00	34.58/13.64/31.43
	D1	0.97	11.69	0.82	0.40	59.47	0.11	50.92	31.44/11.72/28.58
	Mix	0.97	11.1	0.81	0.46	54.66	0.10	53.7	**34.91**/14.36/**31.93**
NROOM	Ref	0.83	3.40	0.46	0.57	23.67	0.07	50.8	-
	Baseline	0.96	14.34	0.80	0.63	34.11	0.10	48.64	**36.38/19.54/31.20**
	D0	0.90	6.15	0.59	0.65	33.95	0.10	49.58	34.64/16.59/28.94
	D1	0.96	16.45	0.84	0.58	34.66	0.10	49.41	33.73/17.27/28.90
	Mix	0.96	17.13	0.81	0.63	38.34	0.11	48.38	35.32/18.69/30.31
XSUM	Ref	0.66	1.05	0.16	0.65	21.1	0.09	59.6	-
	Baseline	0.75	1.61	0.27	0.56	19.20	0.09	66.70	**45.14/22.27/37.25**
	D0	0.72	1.37	0.23	0.66	19.72	0.09	60.45	42.82/19.16/34.15
	D1	0.72	1.44	0.23	0.53	19.96	0.09	62.70	42.33/18.56/33.98
	Mix	0.73	1.51	0.25	0.59	19.60	0.10	62.07	44.72/21.47/36.36

Table 1: Comparison of HYDRASUM's generated summaries using individual decoders (D0 and D1) and their model-derived mixture (Mix). Results show significant differences along multiple dimensions (highlighted in gray), most notably abstractiveness and specificity for CNN and NEWSROOM, and specificity for XSUM.

in Section 4, we show that we can reliably vary abstractiveness between these two decoder levels using their mixture.

How do HYDRASUM decoders learn different style features? Note that we do not introduce constraints or differ the training of the two decoders in any way; this stylistic partitioning naturally emerges. In fact, both decoders are initialized symmetrically, with BART-LARGE. However, the randomly initialized gate $\mathcal{G}$ assigns different weight coefficients to the two decoders in the mixture, and hence their respective contributions to the output probability is different. This ensures that the gradient updates for the two decoders start to differ from the initial stages of the training itself. Eventually, as training progresses, we see that the two decoders learn very different style features characterized by differently learnt weight parameters.[5]

Quality The ROUGE scores of the generated summaries using the entire HYDRASUM model, i.e. Mix, are comparable to the baseline BART models, even outperforming the baseline for CNN (see Table 1). This shows that additional decoders in HYDRASUM does not hurt quality. Notably, the quality of individual decoders is roughly 2 ROUGE points lower than both the Mix strategy. This is expected; individual decoders generate summaries that exhibit "extreme" or contrasting behaviors along style features (shown above). Therefore, they underperform when evaluated on the entire test set containing a diverse set of styles.

Recent work (Fabbri et al., 2021) has shown that ROUGE is insufficient to evaluate summary quality and recommends human evaluation. We report these results in Section 5; they show that HYDRASUM outperforms or is on par with the baseline for all datasets.

3.3 Diversity Evaluation

HYDRASUM provides a straightforward method to sample multiple summaries from its multiple decoders and their combination. Here, we compare the quality of these diverse set of summaries.

Following prior work in diversity evaluation (Vijayakumar et al., 2018), we report the TopK ROUGE metric: the maximum ROUGE (R1/R2/RL) score over a list of K generated summaries for a given input. This gives an upper bound on the benefit that can be derived from diverse summarization by measuring the closeness of the best generated summary to the reference summary. We set K= 5 for our experiments. For HYDRASUM, multiple summaries are generated by varying the summary-level gating probability g (Strategy 3, Section 2.1). We set $g = \{0, .25, .5, .75, 1\}$; here, $g = 0$ and $g = 1$ correspond to summaries generated using D0 and D1 independently. These are compared to K summaries sampled from the

[5]We re-run these experiments with different gate initializations; style partitioning is observed consistently across runs, although the exact degree of partitioning differs slightly.

Dataset	BS + Beam	BS + Top-k	BS + DBS	HS+Beam
CNN	39.10/17.76/35.65	40.29/15.37/36.14	40.62/18.65/37.04	**42.07/19.19/38.32**
NEWSROOM	43.00/24.73/36.98	43.58/22.25/36.27	43.59/24.72/37.27	**45.03/25.59/38.46**
XSUM	50.19/**25.74**/40.86	48.16/21.68/37.98	50.52/25.72/41.06	**51.03**/25.46/**41.18**

Table 2: Diversity performance (TopK R1/R2/RL) of the baseline BART (BS) and HYDRASUM (HS) models.

Dataset	Dec.	Rouge (R1/R2/RL)	2gm	Spec.	Len.
CNN	D0	32.35/10.90/29.29	.48	.34	39.9
	D1	21.63/8.48/20.18	.82	.38	180.7
	D2	33.86/13.23/30.87	.72	.55	56.1
	Mix	**34.30/14.38/31.36**	.82	.48	56.2
NR	D0	31.88/14.71/27.12	.32	.42	32.0
	D1	16.05/6.94/14.39	.36	.49	171.9
	D2	32.43/16.57/27.61	.85	.67	47.9
	Mix	**35.39/18.85/30.37**	.82	.64	38.9
XSUM	D0	31.63/12.21/24.83	.36	.60	44.6
	D1	41.86/17.97/33.22	.22	.54	20.1
	D2	32.33/12.63/25.44	.32	.67	44.1
	Mix	**44.61/20.91/36.17**	.24	.58	19.5

Table 3: Stylistic variation between generated summaries in a 3-decoder HYDRASUM model. Results show higher variation between individual decoders compared to the 2-decoder version.

baseline BART model using the following decoding strategies: beam search, top-k sampling, and diverse beam search (Vijayakumar et al., 2018). Decoding hyperparameters for all settings are in Appendix A.

Table 2 outlines our results. It shows that HYDRASUM substantially outperforms the baseline across all different decoding strategies considered. In fact, **the gain is roughly proportional to the degree of stylistic difference observed in Table 1**; the highest gain (roughly +3 ROUGE points) is reported for CNN, followed by an improvement of +2 ROUGE points for the NEWSROOM dataset.

3.4 Effect of number of decoders

We investigate this by extending our analysis to a 3-decoder variant of HYDRASUM. Table 3 outlines our results. For simpler analysis, we only report 4 metrics: ROUGE, 2-gram overlap, specificity and absolute length.

Similar to the 2-decoder case, the 3 decoders of HYDRASUM learn a mutually-distinct combination of summary styles. In fact, **3-way partitioning allows the model to cover a wider range of summary styles.** For example, the 3-decoder HYDRASUM model partitions along the abstractiveness feature for XSum (D0 and D2 are more extractive compared to D1), while this was not achieved by the 2-decoder variant in Table 1. Similarly, the specificity range for CNN ($.34 - .55$) and NEWSROOM ($.42 - .67$) is higher compared to the 2-decoder variant. Note that some decoders report very poor quality (ROUGE scores). This is expected as these decoders exhibit extreme summary styles (e.g. very long summaries) and therefore suffer on dataset-wide evaluation. However, across all datasets, mixture-decoding outperforms individual decoders. This shows that although the performance of some individual decoders is low, their contribution to the mixture is critical.

3.5 Qualitative Evaluation

Figure 5 shows examples of the style difference between HYDRASUM summaries sampled from individual decoders. In the first example, D1 generates a highly extractive summary whereas D0 generates an abstractive summary with less copying. In the second example, we observe a difference in specificity: D0 summary includes additional details like *Jenson Button*'s profession and his wife's name, compared to the more general summary by D0. HYDRASUM's architecture provides easy access to such stylistically-distinct summary sets.

4 Extreme partitioning

In Section 3, style partitioning was automatically driven by dataset properties. Here, we investigate whether we can explicitly dictate which specific stylistic feature differs between two decoders. Suppose our target feature (denoted by f) is specificity: under this scenario, we want D0 to generate low- and D1 to generate high-specificity summaries. We should also be able to generate multiple mid-specificity summaries by *mixing* these two extreme decoders. In this section, we run experiments on two target features; abstractiveness (measured by 2-gram overlap) and specificity.

Our Method To ensure D0 learns low-f and D1 learns high-f, we carefully control the contribution of each training example to individual decoder's training. Our exact methodology is: (1) First, we pre-process the training data to derive their per-

Input Article	D0 Summary	D1 Summary
Forget gold and oil. Copper prices is the real winner this year. The red metal is up more than 20 percent from its late January low — and that's given one stock a big boost: Freeport-McMoRan. The mining giant is up 40 percent in the same period, but one trader who relies heavily on the technicals and options market, is cautious on the stock, and he warned that the rally could be over. […]	Copper prices are up 20 percent this year, and one miner is up 40 percent. But one trader warns that the rally could be over.	Copper prices is the real winner this year. The red metal is up more than 20 percent from its late January low — and that 's given one stock a big boost.
Jenson Button and his wife Jessica have been robbed at a holiday home in Saint-Tropez. (AAP) - British Formula One star Jenson Button and his model wife Jessica Michibata are believed to have been knocked out with gas during a brazen robbery in which thieves made off with more than A$ 630,000 worth of their possessions. The couple were in a rented villa in the glitzy French coastal resort of Saint-Tropez with friends when the bandits struck. […]	**British Formula One driver** Jenson Button and his wife **Jessica Michibata** have been robbed at their holiday home in Saint-Tropez.	Jenson Button and his model wife have been robbed at their holiday home in Saint-Tropez .

Figure 5: Examples of HYDRASUM summaries from the NEWSROOM dataset. Long extractive spans are underlined, additional details that increase the specificity of summaries are in bold.

	Metric f	Abstractiveness			Specificity		
Model		CNN	NR	XSUM	CNN	NR	XSUM
Prompt-Based	f("*Low*")	.68	.62	.21	.44	.53	.52
	f("*High*")	.83	.84	.24	.53	.76	.69
HYDRA-SUM	f(D0)	.48	.44	.16	.22	.36	.44
	f(D1)	.82	.85	.29	.62	.81	.80

Table 4: Comparison between the extreme partitioning of HYDRASUM and the prompt-based BART models.

Low Spec. Decoder (D0)	High Spec. Decoder (D1)
Two Florida boys are being hailed as local heroes after saving children from a burning mobile home	Isiah Francis, 10, and Jeremiah Grimes, 11, saved two babies from a burning mobile home in Florida.
French prosecutor says he is not aware of any video footage from on board the plane.	French prosecutor says he's not aware of any video footage from on board Germanwings Flight 9525.

Table 5: Example summaries generated using low and high specificity decoders when f = specificity. Extra details in more specific summaries is underlined.

centile scores p based on the f-value of reference summaries (e.g., if f = abstractiveness, we use 2-gram overlap). (2) We derive $K = 5$ partitions of the data based on these percentile scores. For each example, we set its oracle gate probability $g^* \in \{0, 0.25, 0.5, .75, 1\}$ to incorporate information about the percentile split it belongs to. As an example, the bottom 20 percentile of the data (low f) are assigned $g = 0$. (3) Next, instead of using the automatic gating mechanism $\mathcal{G}$ during training, we use the oracle label g^* to derive the mixture coefficients $[1 - g^*, g^*]$ and compute loss as follows:

$$loss = -\sum_i \log\big[(1 - g^*) * P_{\phi_0}(y_i|x, y_{<i}) + g^* * P_{\phi_1}(y_i|x, y_{<i})\big]$$

This allows us to explicitly set the contribution of each training example to different decoders' parameter updates and ensure that D0 and D1 predominantly learn from low- and high-f summaries respectively. Note that the oracles g^* can be defined at the token-, sentence- or summary-level. Since specificity is defined per sentence, we derive individual oracles gates g_t^* for each sentence s_t. For abstractiveness, we use oracle gates derived at the summary-level.

Baseline We compare our model to the popular prompt-based approaches from recent controllable summarization research (He et al., 2022). To emulate the 2 decoder setting of HYDRASUM, we construct 2 prompts "*Low*" and "*High*" to indicate low- and high-f respectively. We divide the training data into two subsets based on their f-values and train models by prepending the prompt to the reference summary. During inference, we sample 2 different summaries using these prompts and compare their f-difference compared to HYDRASUM's extreme partitioning.

Analysis Table 4 outlines our results. For each model, we report f(D0) and f(D1): the average style/feature scores for test summaries generated by D0 and D1 respectively.[6] Our results clearly show that extreme partitioning outperforms the prompt-based baselines. Moreover, it achieves better or more "extreme" partitioning along the target f compared to HYDRASUM decoders in Table 1.

Figure 5 shows examples of generated summaries using the extreme specificity decoders. The high specificity D1 decoder tends to include more details compared to summaries generated using D0.

Can we use HYDRASUM to vary summary styles between these *extremes*? To study this, we gen-

[6]Detailed results with other metrics and examples of *extreme* summaries are included in Appendix D.

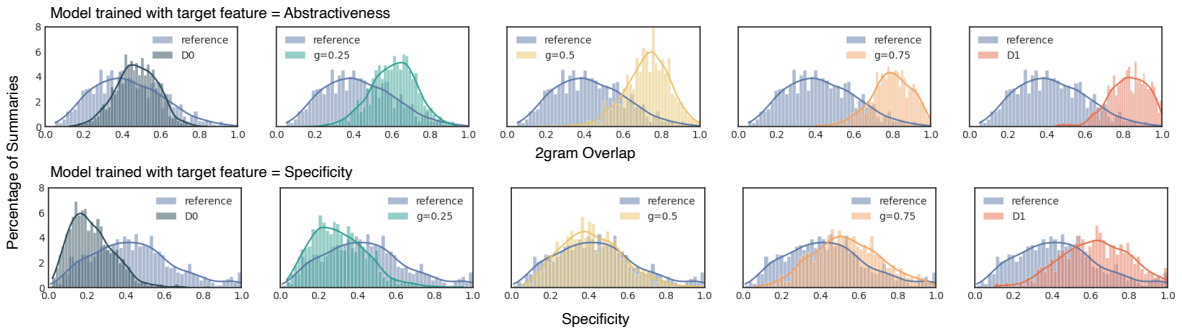


Figure 6: 2gram overlap and specificity of CNN outputs with different values of g under extreme partitioning. The top graphs are from the f = abstractiveness and the bottom are from the f = specificity model. For each, the leftmost graphs correspond to low-f (D0) decoders; the contribution of high-f (D1) increases as we move right. These graphs clearly show that target features can be reliably varied by varying gate probabilities.

erate 5 summaries for each input by varying the gate probabilities: $g = \{0, .25, .5, .75, 1\}$. We plot the 2-gram overlap of CNN summaries for the 5 different gate values for the f = abstractiveness model. Similarly, we plot specificity for the f = specificity model at different gate levels (see Figure 6). Due to space constraints, graphs for NEWSROOM and XSUM are in Appendix D.

For both stylistic features, we observe that the HYDRASUM model shows a gradual increase in average feature scores as the contribution of D1 (high-f decoder) is increased, from 0 contribution in the leftmost graphs to 1 in the rightmost graphs. This shows that HYDRASUM can be used to reliably vary style along a target feature. The graphs also show that our model can sample summaries from a wider area in the generation space compared to baseline models (i.e. compare the 2-gram overlap in Figure 4 with the diversity of overlap in Figure 6).

Can we *mix* decoders of any two separately trained HYDRASUM models? This further tests the flexibility of our models. Here, we run experiments that combine HYDRASUM decoders exhibiting extreme styles along orthogonal features of abstractiveness and specificity (from Section 4), but trained on the same dataset. Choice of such orthogonal styles aids our evaluation by providing a desiderata for generated summaries; if we combine the *highly* extractive and *highly* specific decoders from separate models, we want HYDRASUM to output summaries that follow both these properties.

We conduct this experiment for CNN and NEWSROOM datasets (XSUM is omitted due to low separation along abstractiveness). We target the following pairs, setting gate probability $g = 0.5$: (1) high copy, low specificity, (2) low copy, low specificity, (3) low copy, high specificity and (4) high copy, high specificity. The marginal distribution of each feature for all four combinations is plotted in Figure 7; the left graphs plot 2gram overlap and the right graphs plot specificity. They show that the HYDRASUM summaries generated using a high specificity decoder in the mixture generates more specific summaries on average compared to those using the low specificity decoders. Similar trends are observed for abstractiveness. These results expose potential new use cases of HYDRASUM models, including multi-feature control. We leave further exploration of this capability for future work.

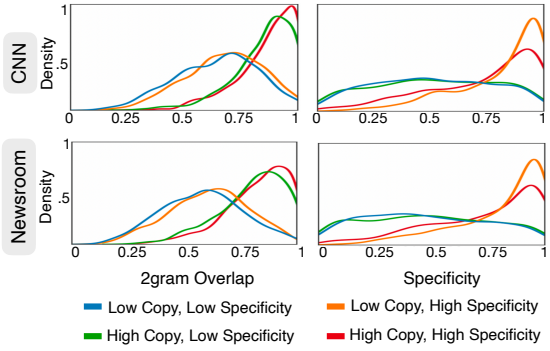


Figure 7: 2-gram overlap and specificity of CNN and NEWSROOM summaries generated using combinations of f = specificity and f = abstractiveness decoders.

5 Human Evaluation

Following prior work (Hashimoto et al., 2019), we conduct human evaluation to measure the quality of generated summaries. For 50 randomly sampled input articles from each dataset, we present MTurk workers with 5 different generated sum-

Data	Model	f = Abs.	f = Spec.
CNN	BS	4.3/4.4/4.2/.83	
	HS D0	**4.4/4.5/4.3/.93**	**4.4**/4.3/4.2/.85
	HS D1	4.3/**4.5/4.3**/.89	**4.4**/4.3/4.1/.87
NROOM	BS	4.2/4.3/4.0/.85	
	HS D0	**4.3/4.4/4.1/.9**	4.1/4.2/3.5/.80
	HS D1	4.2/4.2/4.0/**.9**	4.2/**4.4/4.1**/.85
XSUM	BS	4.3/4.4/**4.2**/0.85	
	HS D0	4.2/4.3/4.1/.89	4.3/4.4/4.1/.81
	HS D1	4.3/**4.5**/4.0/.87	**4.4**/4.4/**4.2/.89**

Table 6: Human-annotated **Relevance/Coherence/ Grammaticality/Factuality** scores for f = abstractiveness and f = specificity HYDRASUM models. We report results for both decoders (D0 and D1) and compare against the baseline BART model.

maries: baseline model summary, D0 and D1 summaries of the f = abstractiveness and f = specificity models. The workers were asked to rate each summary along 4 dimensions: relevance, coherence, grammatically and factuality. For the first 3, we ask for a rating on the 5-point Likert scale. Following Goyal and Durrett (2021), we seek binary labels (factual (1) or non-factual (0)) for factuality annotation. More details and task interface are in Appendix C. We report the average score of all three annotations in Table 6. Across all metrics, we see that the humans score summaries generated by the HYDRASUM models higher than the baseline models. Human annotation results corresponding to the summaries in Table 1 are in Appendix C.

6 Related Work

Prior work on style control in summarization focuses on features like length (Fan et al., 2018a; Song et al., 2021), abstractiveness (Song et al., 2020), etc. It has also been studied for other generation tasks such as paraphrasing and story generation (Wang et al., 2017; Shen et al., 2017; Huang et al., 2019). These methods are over-specialized for the target style and cannot be easily generalized to more features. Recently, GeDi (Krause et al., 2021) proposed using small LMs as generative discriminators for specific attributes (e.g. toxicity) to guide the generation of larger models. Similar class-conditional language models approaches (CC-LMs) have been previously proposed (Keskar et al., 2019; Ficler and Goldberg, 2017) to fine-tune models on specific attributes. Contrary to these, HYDRASUM models can disentangle styles within the task-specific datasets without explicit style annotations, as well as cover the generation space between two 'extreme' styles.

Diverse generation has more widely been studied for other generation tasks, including decoding modifications (Vijayakumar et al., 2018; Kumar et al., 2019), enforcing syntactic diversity (Goyal and Durrett, 2020), or through uninterpretable latent codes (Park et al., 2019; Shao et al., 2019). In this work, we study diversity in style that naturally emerges under standard training and decoding.

7 Conclusion

We propose a new summarization architecture HYDRASUM containing multiple decoders in a mixture-of-experts. Our model automatically separates distinct summary styles, e.g. high or low abstractiveness, different levels of specificity, etc., across different decoders under the standard training regimen. We show that the proposed model is highly flexible; during inference, we can sample from either individual decoders or their mixtures to vary summary features.

8 Limitations

In this paper, we propose a simple modification to existing summarization architectures to disentangle style features. Although this modification is not language-dependant, all our experimentation and analysis is performed only on English language summarization datasets. Furthermore, we only study newswire summaries due to their popularity in summarization research. Therefore, this paper does not provide insights into what style diversity exists in non-English and non-newswire datasets, or whether our findings generalize to these other datasets.

Next, we study style partitioning along a limited number of style dimensions, both due to computational constraints, as well as space constraints in the paper. Due to similar computational constraints, we run all our experiments using the BART model as a case study. While we strongly believe that our conclusions are generalizable to other pre-trained models like PEGASUS, we do not show explicit evidence for this. Note that multiple prior works in summarization have discussed that both BART and PEGASUS exhibit similar high-level trends across various summarization behaviors (Xu et al., 2020; Goodwin et al., 2020).

Acknowledgments

We thank Greg Durrett, Jessy Li and Jiacheng Xu for reviewing an earlier version of this paper and providing valuable feedback. Thanks as well to the Amazon Mechanical Turk workers for participating in the human annotation study and the anonymous reviewers for their helpful comments.

References

Alexander Richard Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*.

Angela Fan, David Grangier, and Michael Auli. 2018a. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018b. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3).

Travis Goodwin, Max Savery, and Dina Demner-Fushman. 2020. Flight of the pegasus? comparing transformers on few-shot and zero-shot multi-document abstractive summarization. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics.

Tanya Goyal, Jiacheng Xu, Junyi Jessy Li, and Greg Durrett. 2022. Training dynamics for text summarization models. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Tatsunori Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. 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)*.

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2022. Ctrlsum: Towards generic controllable text summarization. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*.

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-Volume 70*.

Qiuyuan Huang, Zhe Gan, Asli Celikyilmaz, Dapeng Wu, Jianfeng Wang, and Xiaodong He. 2019. Hierarchically structured reinforcement learning for topically coherent visual story generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using mechanical turk to evaluate open-ended text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *Proceedings of*

the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

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*.

Junyi Jessy Li and Ani Nenkova. 2015. Fast and accurate prediction of sentence specificity. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

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

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*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Sunghyun Park, Seung-won Hwang, Fuxiang Chen, Jaegul Choo, Jung-Woo Ha, Sunghun Kim, and Jinyeong Yim. 2019. Paraphrase diversification using counterfactual debiasing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

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

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.

Kaiqiang Song, Bingqing Wang, Zhe Feng, and Fei Liu. 2021. A new approach to overgenerating and scoring abstractive summaries. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Kaiqiang Song, Bingqing Wang, Zhe Feng, Ren Liu, and Fei Liu. 2020. Controlling the amount of verbatim copying in abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. 2017. Steering output style and topic in neural response generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Jiacheng Xu, Shrey Desai, and Greg Durrett. 2020. Understanding neural abstractive summarization models via uncertainty. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*. PMLR.

A Training Details

Dataset	Training	Dev	Test
CNN	90266	1220	1093
NEWSROOM	329494	35977	36100
XSUM	204045	11332	11334

Table 7: Dataset statistics

We evaluate our models on three datasets: CNN, NEWSROOM and XSUM. Training, development and test dataset sizes for these are listed in Table 7. Note that our experiments (both training and evaluation) are performed on the *mixed* subset of the NEWSROOM dataset. All results and analysis in the paper is reported on the test data.

Table 8 outlines the hyperparameters used for training and inference. For all our experiments, we

For training		For Inference	
Implementation	Huggingface (Wolf et al., 2020)	CNN & NEWSROOM	
Infrastructure	40 GB NVIDIA A100 GPU	Num beams	5
Optimizer	Adam	Length Penalty	2
Optimizer Params	$\beta = (0.9, 0.999), \epsilon = 10^{-8}$	No repetition size	3-grams
Learning Rate Decay	Linear	Min-Length	12
Learning rate	1e-5**	Max Length	200
Weight Decay	0	XSUM	
Maximum Gradient Norm	1	Num beams	6
Batch size	64	Length Penalty	1
Epochs	3	No repetition size	3-grams
Max Input Length	1024 (512 for NEWSROOM)	Min Length	12
Max Output Length	128	Max Length	60

Table 8: Hyperparameters used for fine-tuning and decoding the BART-based summarization models. (**For $f =$ specificity models in Section 4, we set learning rate to 2e-5)

Dataset	m	ROUGE		Overlap		Specificity		Length	
		D0	D1	D0	D1	D0	D1	D0	D1
CNN	6	33.21/13.3/30.21	34.26/13.30/31.21	.79	.63	.42	.43	44.9	54.5
	10	32.04/12.37/29.13	35.20/14.11/32.19	.80	.68	.38	.45	53.8	45.9
NEWSROOM	6	32.32/16.17/27.50	34.92/17.05/29.55	.82	.61	.60	.60	39.5	30.0
	10	33.14/16.56/28.16	34.73/17.10/29.37	.79	.64	.57	.64	33.9	34.6
XSUM	6	42.20/18.70/33.60	42.30/18.70/33.90	.22	.23	.66	.53	20.2	19.8
	10	42.56/19.14/34.10	42.83/19.15/34.24	.24	.23	.64	.56	19.0	20.5

Table 9: Effect of varying the number of shared layers between the 2 decoders of HYDRASUM. Results show that the choice of m does not substantially alter our analysis.

use BART-LARGE as the pre-trained initialization. During inference for HYDRASUM, we incorporate top-k and top-p sampling using values 30 and 0.5 respectively. For top-k decoding using baseline BART model in Table 2, we set $k = 30$. Diverse beam search is run using 2 beam groups and diversity penalty 0.5.

B Effect of different number of shared layers

In order to restrict the number of extra parameters introduced in HYDRASUM, we enforced parameter sharing between the m lower layers of the decoders. We performed our all experiments in Section 3 and 4 by setting $m = 8$. Here, we investigate if the choice of m effects either the partitioning of stylistic features between decoders, or the extent of the observed difference between two decoders along any axis such as abstractiveness, specificity, etc. Experiments are additionally performed using the 2-decoder version of HYDRASUMfor $m = 6, 10$ for all 3 datasets. For simpler analysis, we only report on a subset of the metrics: ROUGE scores (quality), 2 gram overlap (abstractiveness), specificity and absolute length between the summaries generated using individual decoders.

Table 9 outlines the results. Compared to the HYDRASUM model variants with $m = 8$, we notice small differences in style partitioning as well as the absolute difference in style scores between decoders D0 and D1. Most notably, the CNN and NEWSROOM model with 6 shared parameters does not learn to partition across the specificity metric whereas the NEWSROOM model with $m = 6$ does learn to partition along length. These observations are different that those seen for $m = 8, 10$. However, in general, we observe that across all datasets, HYDRASUM decoders behave quite similarly in terms of which features are partitioned, irrespective of the number of shared layers m. This demonstrates that the proposed model architecture is useful for generating diverse summary options, even in cases where a smaller number of extra parameters are allowed.

C Human Evaluation

In section 4, we reported human evaluation study results under extreme partitioning. Here, we expand on the details of the Mechanical Turk task. Figure 10 shows task interface. For each source

article, we asked 3 workers to evaluate 5 different model-generated summaries. For the extreme partitioning setting, these 5 summaries were obtained from (1) Baseline model, (2, 3) D0 and D1 decoders of the f = abstractiveness model, and (4,5) D0 and D1 of the f = abstractiveness model. For each article-summary pair, workers were asked to rate the summaries across 4 metrics: relevance, coherence, grammaticality, and factuality. We follow prior work (Karpinska et al., 2021) and seek annotation for the first 3 on a 5-point Likert scale, with 5 corresponding to highest quality. For factuality, we ask for a binary annotation: 1 for factuality and 0 for non-factual summaries. We report the average scores of the 3 annotators across all 50 articles, for each dataset.

	CNN	NEWSROOM	XSUM
D0	4.3/**4.4**/**4.2**/.86	**4.4**/4.4/4.2/**.92**	**4.3**/4.3/4.2/**.81**
D1	4.3/4.3/4.0/**.89**	4.2/4.4/4.1/.91	4.1/4.4/4.2/.81
Mix	**4.4**/4.3/4.2/.87	**4.4**/**4.5**/**4.3**/.9	4.2/**4.5**/**4.3**/.8
BS	**4.4**/**4.4**/**4.2**/.88	4.3/4.4/4.2/.9	**4.3**/4.4/4.2/.77

Table 10: Comparison of human-annotated **Relevance/Coherence/Grammaticality/Factuality** scores of HYDRASUM models (using individual decoders D0 and D1, and their mixture) and baseline BART (BS).

Next, we conducted an analogous study for our original training setting, corresponding to the standard training regimen. For this, we asked workers to rate the quality of 4 different summaries per article (1) baseline model, (2, 3) D0 and D1 of HYDRASUM model, and (4) Mix strategy of HYDRASUM model. Again, we ask ratings for 50 randomly sampled articles (note that these articles are different from the ones annotated in the baseline setting, and therefore, baseline model results may differ). Table 10 outlines the results. The results show that the HYDRASUM model performs on par with the baseline model along all quality dimensions measured, even outperforming it in terms of factuality for both NEWSROOM and XSUM. This agrees with our results from Table 1 which similarly shows that both the baseline and HYDRASUM model summaries have similar quality.

D Extreme Partitioning - Additional Results

In Section 4, we reported the style scores of the different models under our *extreme* partitioning scenario. Table 4 outlined a brief summary of results

f	Dec.	Quality	Summary Styles		
		ROUGE	Ov.	Sp.	Len
CNN					
Abs.	D0	35.00/12.93/31.84	.48†	.42	48.8
	D1	34.66/14.45/31.78	.82†	.42	46.2
Spec.	D0	33.64/12.74/30.70	.72	.22†	48.9
	D1	34.40/13.35/31.18	.69	.62†	49.7
NEWSROOM					
Abs.	D0	32.56/13.98/26.68	.44†	.65	35.8
	D1	35.04/18.53/30.17	.85†	.59	33.9
Spec.	D0	31.62/14.80/27.11	.67	.36†	27.0
	D1	34.20/17.26/28.74	.73	.81†	38.4
XSUM					
Abs.	D0	42.45/19.00/34.35	.16†	.58	19.2
	D1	43.52/19.79/35.05	.29†	.57	19.5
Spec.	D0	41.84/18.55/33.86	.22	.44†	18.2
	D1	41.72/18.14/33.11	.22	.80†	21.8

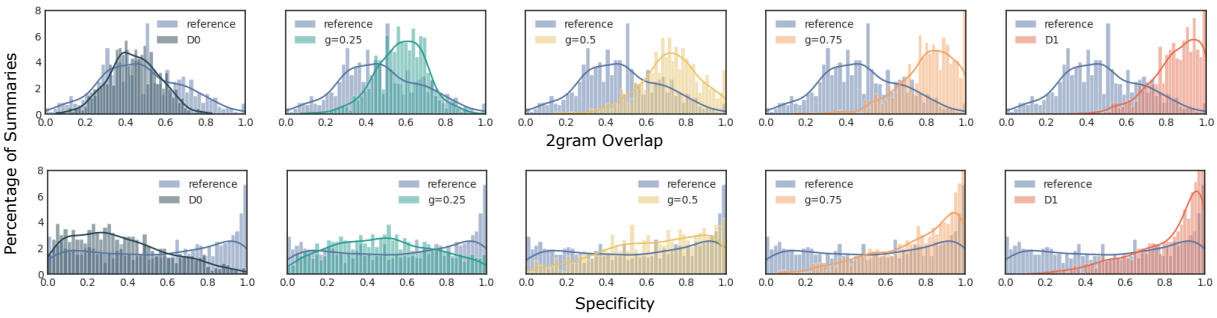
Table 11: Performance of *extreme* partitioned HYDRASUM models. Compared to Table 1, we observe higher variation in style between D0 and D1 along the target dimension (indicated with †)

for models trained on the three datasets. Here, we provide the entire set of results, see Table 11. In addition to the metrics reported in the main paper, we include ROUGE scores of individual decoders D0 and D1 for both $f \in$ { abstractiveness, specificity} models. Moreover, other style metrics (in addition to the target f of each model) are also included for each model and dataset pair (2-gram overlap, specificity and length). Table 11 outlines the results. In general, we observe that HYDRASUM models are able to enforce diverse generation along the target feature f, while limiting the stylistic variance along other features between D0 and D1. Figure 5 includes examples of low- and high-specificity summaries generated using the f = specificity model.

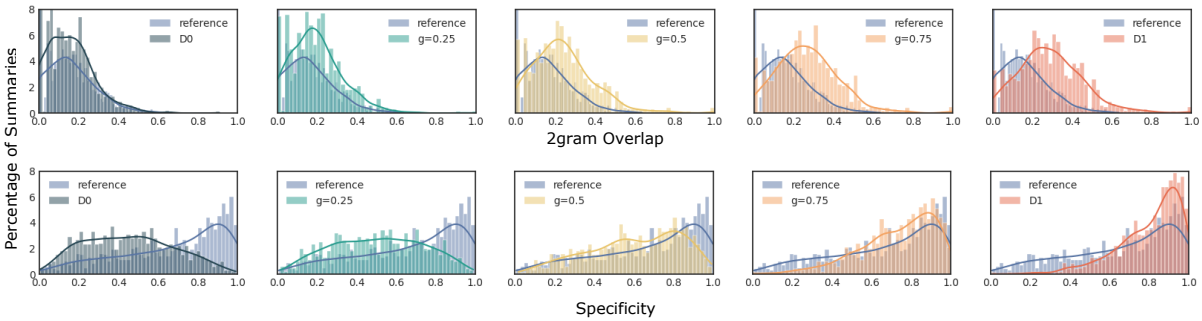
Finally, in Figure 8, we include graphs that show the distributions of 2 gram overlap and specificity for the f = abstractiveness (top row) and f = specificity (bottom row) models respectively, for datasets NEWSROOM and XSUM models. The corresponding graphs for CNN are included in the main body of the paper (section 4).

E Combining multi-feature decoders

Figure 8 shows an example of summaries generated using a combination of *extreme* decoders corresponding to orthogonal features for the NEWSROOM dataset. We 4 generate summaries by using a distinct combination of extractive/abstractive



(a) NEWSROOM



(b) XSUM

Figure 8: 2gram overlap and specificity of NEWSROOM and XSUM summaries generated using different values of g. The graphs show that properties like abstractiveness and specificity can be varied by sampling from a mixture of the 2 decoders corresponding to the chosen style.

and general/specific decoders from different single-feature controlled models. The figure shows the input article and these generated summaries: we see that these summary follow the style specifications of the two decoders used to construct them. Interestingly, for the High Copy, Low specificity summary, we see that the model replaces *Lyft* with *ride-sharing company* and *VanderSaden* with *former executive* from an exact copied sentence from the input, to both follow high copy and low specificity targets as faithfully as possible. In general, we found summary generation including a low specificity decoder tougher to control (here, the Low copy, Low Specificity summary follows similar strategy to the High Copy, Low Specificity summary). This is also evidenced by specificity distributions in Figures 8 which show much higher variation for D0 (i.e. low specificity decoder) for the specificity controlled model. Similar trends are seen in Figure 7.

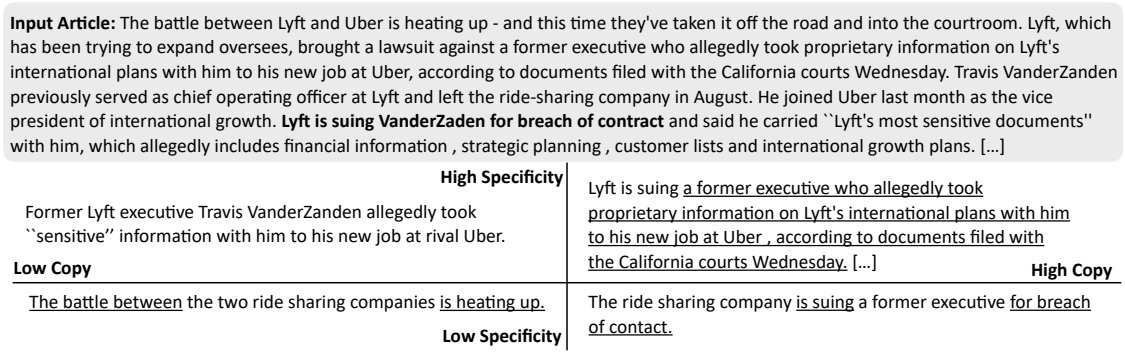


Figure 9: Examples of summaries generated by combining HYDRASUM decoders from different models and corresponding to different extreme styles.

Figure 10: Interface of the Mechanical Turk Task