Boosting Summarization with Normalizing Flows and Aggressive Training

Yu Yang & Xiaotong Shen

University of Minnesota

UNIVERSITY OF MINNESOTA

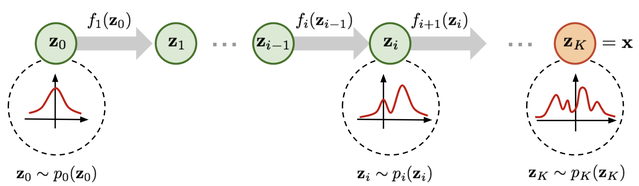
Motivation

Variational Models introduce uncertainty by learning a probability distribution over the latent variables. These models can generate smoother output spaces, more diverse summaries, and facilitate improved capture of semantic information.

Commonly used models typically utilize basic Gaussians, which are rigid in capturing latent intricacies. Moreover, numerous variational models encounter the issue of posterior collapse. In addressing this, we suggest the utilization of **Normalizing Flows** alongside **Aggressive Training**.

Background

Normalizing Flows[a]



$$\mathbf{z}_i = f_i(\mathbf{z}_{i-1}) \Leftrightarrow \mathbf{z}_{i-1} = f_i^{-1}(\mathbf{z}_i); \quad \log p_K(\mathbf{z}_K) = \log p_0(\mathbf{z}_0) - \sum_{i=1}^{K} \log \left| \det \frac{df_i}{dz_{i-1}} \right|$$

Model Details

NF Latent Module (in purple)

1. Pass average embedding $\overline{x}$ to an inference network $q_0 : \overline{x} \mapsto (\mu_0, \sigma_0)$
2. Sample $z_0 \sim N(\mu_0, \text{diag}(\sigma_0^2))$
3. Apply K layers of normalizing flows transformation: $z_K = f_K \circ \cdots \circ f_1(z_0)$

Refined Gate Mechanism (in orange)

Mitigates saturation and allows for better gradient flow [1].

$$\begin{cases} z'_K = W^z z_K \\ f_j = \delta\left(W^f [h_j; z'_K]\right) \\ h'_j = (1 - f_j) \cdot h_j + f_j \cdot z'_K \end{cases} \Rightarrow \begin{cases} z'_K = W^z z_K \\ f_j = \delta\left(W^f [h_j; z'_K]\right) \\ r_j = \delta\left(W^r [h_j; z'_K]\right) \\ g_j = f_j + f_j(1 - f_j)(2r_j - 1) \\ h'_j = (1 - g_j) \cdot h_j + g_j \cdot z'_K \end{cases}$$

Controlled Alternate Aggressive Training (CAAT)

Mitigate the posterior collapse problem:

1. Alternately update variational parameters and entire parameters for n_{agg} steps
2. Train all parameters jointly for the remainder of the training.

Objective Function

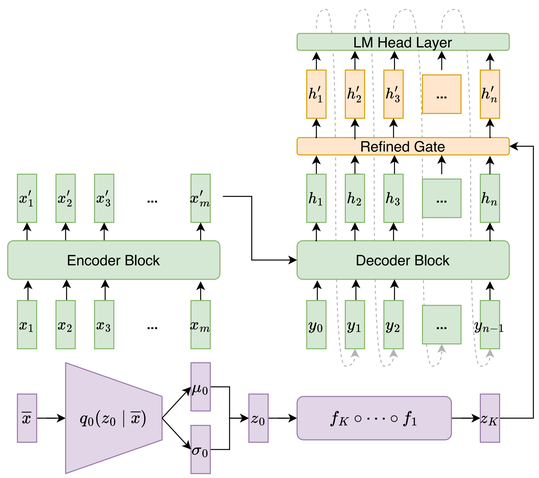
$$\text{ELBO}_{\text{NF-VED}} = \mathbb{E}_{q_0(z_0)}\left[\log p(y \mid x, z_K) + \log p(z_K \mid x)\right] - \mathbb{E}_{q_0(z_0)}\left[\log q_0(z_0) - \sum_{k=1}^{K} \log \left|\det J_{f_k}(z_{k-1})\right|\right], \quad (1)$$

$$\mathcal{L} = -\sum_{j=1}^{n} \log p\left(y_j \mid \{x_i\}_{i=1}^{m}, z_K, y_{<j}\right) - \log p(z_K \mid x) + \log q_0(z_0) - \sum_{k=1}^{K} \log \left|\det J_{f_k}(z_{k-1})\right| \quad (2)$$

where q_0 is the probability density function for z_0.

[a] Image source: `https://lilianweng.github.io/posts/2018-10-13-flow-models/`.

Model: FlowSUM



Quantitative Analyses

Model	R1	R2	RL
PG+Cov	39.53	17.28	36.38
BERT2BERT	41.28	18.69	38.09
BERTSUM	42.13	19.60	39.18
BART	44.16	21.28	40.90
PEGASUS	44.17	21.47	41.11
VHTM	40.57	18.05	37.18
TAS	44.38	21.19	41.33
PEGASUS+NTM	44.52	**21.95**	41.39
VEDSUM (BERT2BERT)	40.89	18.28	37.95
FlowSUM (BERT2BERT)	41.51	18.81	38.56
VEDSUM (BART)	44.36	21.09	41.37
FlowSUM (BART)	**44.64**	21.36	**41.65**

Table 1. Comparison with baselines on CNN/DM.

Model	ROUGE 1/2/L ↑	BERTScore ↑	rep-w ↓	Length
CNN/DM				
BART	44.16/21.28/40.90	89.40	**8.31**	84.11
VEDSUM	44.34/21.09/41.37	89.20	8.43	88.63
FlowSUM	**44.64**/**21.36**/**41.65**	**89.46**	8.43	92.24
Multi-News				
BART	42.56/15.34/36.67	86.69	**9.76**	133.42
VEDSUM	43.91/16.68/38.10	87.04	9.95	128.79
FlowSUM	**44.42**/**17.01**/**38.36**	**87.09**	9.91	128.87
arXiv				
BART	42.55/15.92/37.89	85.35	17.23	130.68
VEDSUM	43.05/**16.34**/38.26	85.44	16.63	130.92
FlowSUM	**43.11**/16.26/**38.31**	**85.45**	**16.55**	132.88
PubMed				
BART	41.57/16.72/36.94	84.65	13.26	136.10
VEDSUM	44.21/19.20/39.32	85.07	12.76	138.70
FlowSUM	**44.55**/**19.50**/**39.59**	**85.16**	**12.59**	138.09

Table 2. Comparison of BART, VEDSUM, and FlowSUM on long-summary benchmarks.

Ablation Studies

Model	ROUGE 1/2/L ↑	BERTScore ↑	rep-w ↓
BART	42.56/15.35/36.67	86.69	**9.76**
VEDSUM	43.91/16.68/38.10	87.04	9.95
FlowSUM (Planar)	43.85/16.61/37.97	87.03	10.04
FlowSUM (Radial)	43.84/16.68/37.98	87.04	9.92
FlowSUM (Sylvester)	44.18/16.71/38.15	87.08	9.80
FlowSUM (RealNVP)	44.19/16.64/38.15	87.05	9.81
FlowSUM (IAF)	**44.42**/**17.01**/**38.36**	**87.09**	9.91
FlowSUM (RLNSF)	44.25/16.86/38.14	87.06	9.80
FlowSUM (RQNSF)	44.31/16.98/38.27	87.07	9.91

Table 3. Effect of NF Types on Multi-News.

Model	Training	R1	R2	RL	KL Div
VEDSUM	standard	43.91	16.68	38.10	0.0117
VEDSUM	β_C-VAE	43.78	16.54	37.96	0.0082
FlowSUM (IAF)	standard	43.87	16.62	37.97	3.9146
FlowSUM (IAF)	β_C-VAE	43.81	16.58	37.91	3.9128
FlowSUM (IAF)	CAAT	44.30	17.03	38.22	2.1108
FlowSUM (RQNSF)	standard	44.18	16.76	38.18	127.8106
FlowSUM (RQNSF)	β_C-VAE	44.18	16.76	38.18	127.8106
FlowSUM (RQNSF)	CAAT	44.31	16.98	38.27	107.0794

Table 4. Effect of Training Strategies.

FlowSUM on Knowledge Distillation

FlowSUM helps with knowledge distillation with both SFT and PL [2].

Model	ROUGE ↑ 1/2/L	BERT-Score ↑	Length	# Params (MM)	Inference Time (MS) ↓
dBART-6-6					
dBART-6-6	42.78/20.24/39.72	88.98	67.42	230	170.5
FlowSUM	43.41/20.33/40.41	89.18	91.25	238	234.9
FlowSUM-PLKD	**43.70**/**20.71**/**40.73**	**89.24**	91.10	238	239.7
dBART-12-3					
dBART-12-3	43.39/20.57/40.44	89.20	85.48	255	199.6
FlowSUM	43.53/20.61/40.59	89.28	83.74	263	190.7
FlowSUM-PLKD	**44.05**/**21.06**/**41.07**	**89.37**	84.48	263	200.4

Table 5. Knowledge Distillation on DistilBART on CNN/DM.

References

[1] Albert Gu, Caglar Gulcehre, Thomas Paine, Matt Hoffman, and Razvan Pascanu. Improving the gating mechanism of recurrent neural networks. In *International Conference on Machine Learning*, pages 3800–3809. PMLR, 2020.

[2] Sam Shleifer and Alexander M Rush. Pre-trained summarization distillation. *arXiv preprint arXiv:2010.13002*, 2020.