A Performance/Controllability trade-off

The trade-off between performance and interpretability has been a long-standing problem in feature selection (Jackson, 1998; Haury et al., 2011). The trade-off exists because it is usually very difficult to accurately find the exact features needed to make the prediction. Safely keeping more features will almost always lead to better performance. Some models do succeed in achieving superior performance by selecting only a subset of the input. However, they mostly still target at the recall of the selection (Hsu et al., 2018; Chen and Bansal, 2018; Shen et al., 2018a), i.e., to select all possible content that might help predict the target. The final selected contents reduce some most useful information from the source, but they still contain many redundant contents (same like our VRS-($\epsilon = 0$) as in Table 6 and 5). This makes them unsuitable for controllable content selection. In text generation, a recent work from Moryossef et al. (2019) shows they could control the contents by integrating a symbolic selector into the neural network. However, their selector is tailored by some rules only for the RDF triples. Moreover, even based on their fine-tuned selector, the fluency they observe is still slightly worse than a standard seq2seq.

We assume the content selector is the major bottle if we want a model that can achieve controllability without sacrificing the performance. We can clearly observe in Table 6 that the performance drop in Wikibio is marginal compared with Gigaword. The reason should be that the selection on Wikibio is much easier than Gigaword. The biography of a person almost always follow some simple patterns, like name, birthday and profession, but for news headlines, it can contain information with various focuses. In our two tasks, due to the independence assumption we made on β_i and the model capacity limit, the content selector cannot fully fit the true selecting distribution, so the trade-off is necessary. Improving the selector with SOTA sequence labelling models like Bert (Devlin et al., 2019) would be worth trying.

There are also other ways to improve. For example, we could learn a ranker to help us choose the best contents (Stent et al., 2004). Or we could manually define some matching rules to help rank the selection (Cornia et al., 2018). In Table 2, we show the VRS model achieves very high metric scores based on an oracle ranker, so learning a ranker should be able to improve the performance straightforwardly.

Source: The sri lankan government on Wednesday announced the closure of government schools with immediate effect as a military campain against tamil separatists escalated in the north of the country.
Reference: sri lanka closes schools as war escalates .
b1: sri lanka shuts schools as war escalates .
b2: sri lanka closes schools as violence escalates .
b3: sri lanka shuts schools as fighting escalates .
b4: sri lanka closes schools as offensive expands .
b5: sri lanka closes schools as war continues .

Figure 4: Posterior inference example. Highlighted words are selected contents according to the posterior distribution $q_\phi(\beta|X, Y)$. b1-b5 are decoded by fixing the selected contents.

B Example from Wikibio

To see how we can manually control the content selection, Figure 5 shows an example from Wikibio, the model is mostly able to form a proper sentence covering all selected information. If the selector assigns very high probability to select some content and we force to remove it, the resulting text could be unnatual (as in summary 4 in Figure 5 because the model has seen very few text without containing the birthday information in the training corpus). However, thanks to the diversity of the content selector as shown in the previous section, it is able to handle most combinatorial patterns of content selection.

C Posterior inference

Figure 4 further provides an example of how we can perform posterior inference given a provided text. Our model is able to infer which source contents are covered in the given summary. With the inferred selection, we can sample multiple paraphrases describing the same contents. As seen in Table 6 and 5, the metric scores are remarkably high when decoding from the posterior inferred selections (last three rows), suggesting the posterior distribution is well trained. The posterior inference part could be beneficial for other tasks like content transfer among text (Wang et al., 2019; Prabhumoye et al., 2019). The described source contents can be first predicted with the posterior inference, then transferred to a new text.

Personal information	
Full name	Dillon Douglas Sheppard
Date of birth	27 February 1979 (age 39)
Place of birth	Durban, South Africa
Height	1.80 m (5 ft 11 in)
Playing position	Left-winger
Club information	
Current team	Bidvest Wits
Number	29
...	

Selected Content: Dillon Douglas Sheppard, 27 February 1979, Left-winger
Summary 1: dillon douglas sheppard (born 27 february 1979) is a football (soccer) left-winger .

Selected Content: Dillon Douglas Sheppard, 27 February 1979, South Africa, Left-winger
Summary 2: dillon douglas sheppard (born 27 february 1979) is a south african football (soccer) left-winger .

Selected Content: Dillon Douglas Sheppard, 27 February 1979, South Africa, Left-winger, Bidvest Wits
Summary 3: dillon douglas sheppard (born 27 february 1979) is a south african football (soccer) left-winger who plays for bidvest wits.

Selected Content: Dillon Douglas Sheppard, Left-winger, Bidvest Wits
Summary 4: dillon douglas sheppard (born) is a football (soccer) left-winger who plays for bidvest wits.

Figure 5: Example of content selection in Wikibio. Summary 4 in unnatural because we force to remove the birthday content which the selector assigns very high probability to select.

	Gigaword	Wikibio
Vocabulary	Built with byte-pair segmentation with size 30k	Built by keeping the most frequent 20k tokens
Word Embedding	Size 300. Initialized with Glove (Pennington et al., 2014). OOVs are randomly initialized from a normal distribution	
Inputs	Sequence of source word embeddings	Sequence of concatenation of source table field, value, position and reverse position embeddings (Lebret et al., 2016)
Source Encoder	Single-layer Bi-LSTM with hidden size 512	Single-layer Bi-LSTM with hidden size 500
Target Decoder	Single-layer LSTM with hidden size 512	Single-layer LSTM with hidden size 500
Drop out rate	0.3 for both encoder and decoder	
Decoding Method	Beam search with beam size 5	Greedy decoding. UNK words are replaced with the most attended source token as in Liu et al. (2018)
Mini-batch size	256	128
Optimizer	Adam, $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$, weight decay $= 1.2 \times 10^{-6}$, gradient clipping in [-5,5]	
Initial Learning Rate	0.0005	
Prior Selector	$\mathbf{B}(\gamma_i) = \sigma(\text{MLP}(h_i))$. MLP is a multi-layer perceptron. h_i comes from the encoder hidden state	
Posterior Selector	$q_\phi(\beta_i \vert X, Y) = \sigma(\text{MLP}([h_i \circ e(y)]))$. $\circ$ means concatenation	

Table 7: Detailed settings of our experiment. $e(y)$ in the posterior selector is an encoded representation of the ground-truth text. We use a bi-LSTM encoder. The last hidden state is treated as the representation for the text.