Posterior Control of Blackbox Generation

Xiang Lisa Li
Department of Computer Science
Johns Hopkins University
xli150@jhu.edu

Alexander M. Rush
Department of Computer Science
Cornell Tech
arush@cornell.edu

Abstract

Text generation often requires high-precision output that obeys task-specific rules. This fine-grained control is difficult to enforce with off-the-shelf deep learning models. In this work, we consider augmenting neural generation models with discrete control states learned through a structured latent-variable approach. Under this formulation, task-specific knowledge can be encoded through a range of rich, posterior constraints that are effectively trained into the model. This approach allows users to ground internal model decisions based on prior knowledge, without sacrificing the representational power of neural generative models. Experiments consider applications of this approach for text generation. We find that this method improves over standard benchmarks, while also providing fine-grained control.

1 Introduction

A core challenge in using deep learning for NLP is developing methods that allow for controlled output while maintaining the broad coverage of data-driven methods. While this issue is less problematic in classification tasks, it has hampered the deployment of systems for conditional natural language generation (NLG), where users often need to control output through task-specific knowledge or plans. While there have been significant improvements in generation quality from automatic systems (Mei et al., 2016; Dusek and Jurcicek, 2016; Lebret et al., 2016b), these methods are still far from being able to produce controlled output (Wiseman et al., 2017). Recent state-of-the-art system have even begun to utilize manual control through rule-based planning modules (Moryossef et al., 2019; Puduppully et al., 2019).

Consider the case of encoder-decoder models for generation, built with RNNs or transformers. These models generate fluent output and provide flexible representations of their conditioning. Unfortunately, auto-regressive decoders are also globally dependent, which makes it challenging to incorporate domain constraints.

Research into controllable deep models aims to circumvent the all-or-nothing dependency tradeoff of encoder-decoder systems and expose explicit higher-level decisions. One line of research has looked at global control states that represent sentence-level properties for the full decoder. For example, Hu et al. (2017) uses generative adversarial networks where the attributes of the text (e.g., sentiment, tense) are exposed. Another line of research exposes fine-level properties, such as phrase type, but requires factoring the decoder to expose local decisions, e.g. Wiseman et al. (2018).

This work proposes a method for augmenting any neural decoder architecture to incorporate fine-grained control states. The approach first modifies training to incorporate structured latent control variables. Then, training constraints are added to anchor the state values to problem-specific knowledge. At test time, the control states can be ignored or utilized as grounding for test-time constraints. Technically, the approach builds on recent advances in structured amortized variational inference to enforce additional constraints on the learned distribution. These constraints are enforced through efficient structured posterior calculations and do not hamper modeling power.

We demonstrate that the method can improve accuracy and control, while utilizing a range of different posterior constraints. In particular on two large-scale data-to-text generation datasets, E2E (Novikova et al., 2017) and WikiBio (Lebret et al., 2016a), our method increases the performance of benchmark systems while also producing outputs that respect the grounded control states. Our code is available at https://github.com/XiangLi1999/

PosteriorControl-NLG.

2 Control States for Blackbox Generation

Consider a conditional generation setting where the input consists of an arbitrary context x and the output $y_{1:T}$ is a sequence of target tokens. We are interested in modeling latent fine-grained, discrete control states $z = z_{1:T}$ each with a label in $\mathcal{C}$. We assume that these states are weakly-supervised at training through problem-specific constraints. The goal is to induce a model of $p(y \mid x) = \sum_z p(y, z \mid x)$. Concretely, our experiments will focus on a data-to-text generation problem where x corresponds to a table of data, and $y_{1:T}$ is a textual description. We hope to induce control states z that indicate which table fields are being described, and our weak supervision corresponds to indicators of known alignments.

We assume the generative model is a blackbox auto-regressive decoder that produces both y and z. Define this general model as:

$$p_\theta(y, z \mid x) = \prod_{t=1}^{T} \begin{array}{l} p_\theta(y_t \mid x, y_{<t}, z_{\le t}) \cdot \\ p_\theta(z_t \mid x, y_{<t}, z_{<t}) \end{array}$$

For a neural decoder, where $h_t(y_{1:t-1}, z_{1:t-1})$ is the hidden state at time-step t, we might generate the latent class $z_t \in \mathcal{C}$ and next token y_t as,

$$\begin{aligned} p_\theta(z_t \mid z_{<t}, y_{<t}) &= \mathrm{softmax}(W_0 h_t + b_0) \\ p_\theta(y_t \mid z_{\le t}, y_{<t}) &= \mathrm{softmax}(W_1[h_t, g_\theta(z_t)] + b_1) \end{aligned}$$

Here g_θ is a parameterized embedding function and W, b are model parameters from θ. The log-likelihood of the model is given by $\mathcal{L}(\theta) = \log p_\theta(y \mid x)$.

The key latent term of interest is the posterior distribution $p_\theta(z \mid x, y)$, i.e. the probability of over state sequences for a known output. The decoder parameterization makes this distribution intractable to compute in general. We instead use variational inference to define a parameterized variational posterior distribution, $q_\phi(z \mid x, y)$, from a preselected family of possible distributions $\mathcal{Q}$.[1] To fit the model parameters θ, we utilize the evidence lower bound (for any variational parameters ϕ),

$$\begin{aligned} &\mathcal{L}(\theta) \ge \mathrm{ELBO}(\theta, \phi) \\ &= \mathbb{E}_{z \sim q_\phi(z|x,y)}[\log p_\theta(y, z \mid x)] + \mathrm{H}[q_\phi(z \mid x, y)] \end{aligned}$$

Several recent works have shown methods for effectively fitting neural models with structured variational inference (Johnson et al., 2016; Krishnan et al., 2017; Kim et al., 2019). We therefore use these techniques as a backbone for enforcing problem-specific control. See §4 for a full description of the variational family used.

3 Posterior Regularization of Control States

Posterior regularization (PR) is an approach for enforcing soft constraints on the posterior distribution of generative models (Ganchev et al., 2010). Our goal is to utilize these soft constraints to enforce problem specific weak supervision. Traditionally PR uses linear constraints which in the special case of expectation maximization for exponential families leads to convenient closed-form training updates. As this method does not apply to neural generative models, we resort to gradient-based methods. In this section, we develop a form of posterior regularization that accommodates the neural variational setting.

Starting with the log-likelihood objective, $\mathcal{L}(\theta)$, PR aims to add distributional constraints on the posterior. These soft constraints are expressed as a distributional penalty, $R_p(x, y) \ge 0$. For example, if we have partial information that a specific control state takes on label c we can add a constraint $R_p(x, y) = 1 - p(z_t = c \mid x, y)$. We might also consider other distributional properties, for instance penalizing the entropy of a specific posterior marginal, $R_p(x, y) = \mathrm{H}_{z'}(z_t = z' \mid x, y)$. See §5 for more constraint examples.

PR uses these soft constraints to regularize the model. Ideally we would penalize the posterior directly, but as noted above, computing this term in a blackbox model is intractable. We therefore follow Ganchev et al. (2010) and use a relaxed version with a surrogate posterior $q_\phi(z \mid x, y)$,

$$\begin{aligned} \mathcal{L}_{PR}(\theta) \;=\; & \mathcal{L}(\theta) - \\ & \min_\phi [\mathrm{KL}[q_\phi \,||\, p_\theta(z \mid x, y)] + \lambda R_{q_\phi}(x, y)] \end{aligned} \tag{1}$$

We can write this in terms of a variational lower-bound on the relaxed PR objective.

$$\begin{aligned} \mathcal{L}_{PR}(\theta) \;\ge\; & \mathrm{PRLBO}(\theta, \phi) = \mathcal{L}(\theta) - \\ & [\mathrm{KL}[q_\phi \,||\, p_\theta(z \mid x, y)] + \lambda R_{q_\phi}(x, y)] \end{aligned} \tag{2}$$

This allows us to relate the q in the PRLBO to the variational posterior in the ELBO simply by

[1] Since our family is over a combinatorial set of $z_{1:T}$, this corresponds to a *structured* variational inference setting.

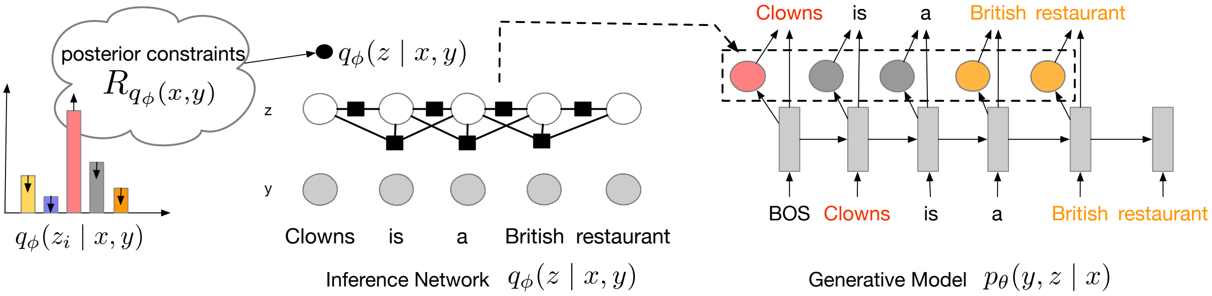

posterior constraints
$R_{q_\phi(x,y)}$
$q_\phi(z_i \mid x, y)$
$q_\phi(z \mid x, y)$
z
y
Clowns is a British restaurant
Inference Network $q_\phi(z \mid x, y)$
Clowns is a British restaurant
BOS Clowns is a British restaurant
Generative Model $p_\theta(y, z \mid x)$

Figure 1: Model training. Assumes we are given conditioning x (not shown) and output sentence y. (Middle) An inference network ϕ is used to parameterize a structured segmental conditional random field $q_\phi(z \mid x, y)$ over control states z. (Right) Sample from q_ϕ (colored circles) is used to provide control state labels for a blackbox generation model $p_\theta(y, z \mid x)$. (Left) To ground the control states to represent problem-specific meaning, posterior regularization is used to enforce distributional constraints through penalties $R_q(x, y)$. The whole system is optimized end-to-end to learn latent properties of the final output tokens.

expanding the KL and rearranging terms,

$$\text{PRLBO}(\theta, \phi) \quad = \quad \text{ELBO}(\theta, \phi) - \lambda R_{q_\phi}(x, y)$$

To train, we jointly maximize over both terms in the PRLBO: the model parameters θ and the variational parameters ϕ (which tightens the bounds). Following standard practice, we use an amortized inference network, i.e. a variational autoencoder (Kingma and Welling, 2014; Mnih and Gregor, 2014; Rezende et al., 2014), to define ϕ.

4 Structured Variational Family for Segmental Generation

We now discuss how to efficiently compute the PRLBO under a structured variational family.

$$\text{PRLBO} = \underbrace{\mathbb{E}_{z\sim q_\phi}[\log p_\theta]}_{(1)} + \underbrace{\text{H}[q_\phi]}_{(2)} - \underbrace{\lambda R_{q_\phi}(x, y)}_{(3)}$$

We need a $q_\phi(z \mid x, y)$ for which we can efficiently (1) take samples, (2) compute entropy, and (3) compute the distributional penalties. This motivates the use of a factored conditional random field (CRF), defined by a potential function $\phi(x, y, z)$. At training time, x, y are observed and z is the latent variable that denotes the control states. We then specify a variational posterior distribution: $q_\phi(z \mid x, y) = \frac{\phi(x,y,z)}{\sum_{z'} \phi(x,y,z')}$.

In this work, we focus on the semi-Markov CRF (Gales and Young, 1993; Sarawagi and Cohen, 2005), a common CRF family used in generation (Wiseman et al., 2018). It divides tokens into segmental spans, which are useful for generating entity mentions and commonly used phrases. This model divides the potential function into three parts: the **emission** potential for a span of tokens given a state, denoted as $\phi_{(e)}$; the **transition** potential between states, $\phi_{(t)}$; and the **length** potential of span length given a state, $\phi_{(l)}$. Suppose our control states define a span from i (inclusive) to j (exclusive) labeled by c, we denote it as $z_{i:j} = c$. The potential function of a labeled sequence is defined:

Algorithm 1: Generic Semi-Markov Algorithm.

```
Given $\phi$ and generic semiring $(\oplus, \otimes, \mathbf{0}, \mathbb{1})$
Set $\beta_T(c) = \mathbb{1} \; \forall c \in \mathcal{C}$
for $i = T - 1, \dots, 0$ do
    for $c \in \mathcal{C}$ do
        $\beta'_i(c) = \bigoplus_{d=1}^{\min(L,T-i)} \beta_{i+d}(c) \otimes \phi_{(l)}(d) \otimes \phi_{(e)}(x, y_{i,i+d}, c)$
    for $c \in \mathcal{C}$ do
        $\beta_i(c) = \bigoplus_{c' \in \mathcal{C}} \beta'_i(c') \otimes \phi_{(t)}(c, c')$
return $Z = \bigoplus_{c \in \mathcal{C}} \beta'_0(c) \otimes \phi_{(t)}(0, c)$
```

$$\phi(x, y, z) = \prod_{i<j<k} \phi_{(t)}(z_{i:j}, z_{j:k}) \cdot \phi_{(l)}(j - i) \cdot \phi_{(e)}(x, y_{i:j}, z_{i:j}) \quad (3)$$

For computational efficiency, we restrict all segment length to be $\leq L$.[2]

With this model, we can use the forward-backward algorithm for all required inferences: exact sampling, computing partition function, entropy, and posterior marginals $q_\phi(z_{i:j} = c \mid x, y)$, useful for term (3). In Algorithm 1, we give a

[2] The time complexity to compute the posterior moments of the full semi-Markov CRF is $O(|\mathcal{C}|^2 nL)$.

One-to-One Name	 Penalty	**One-to-Many** Name	 Penalty
Inclusion	For $(i, j, f) \in A(x, y)$, $R_q = 1 - q(z_{i:j} = \sigma(f) \mid x, y)$	Sparsity	For $f \in \mathcal{F}$, $R_q = \text{H}[\sigma(c \mid f)]$
Exclusion	For $f \in x$ and $(i, j, f) \notin A(x, y)$, $R_q = q(z_{i:j} = \sigma(f) \mid x, y)$	Fit	For $(i, j, f) \in A(x, y)$ $R_q = \text{H}[\sigma(c \mid f), q(z_{i:j} \mid x, y)]$
Coverage	For $f \in \mathcal{F}$, $R_q = \lvert \sum_{i<j} q(z_{i:j} = \sigma(f) \mid x, y) - \mathbb{1}(f \in x) \rvert$	Diversity	Let $p_{\text{agg}}(\hat{z}) \propto \sum_{t=1}^{T} q(z_t = \hat{z} \mid x, y)$ $R_q = \text{H}[\text{Unif}(\hat{z})] - \text{H}[p_{\text{agg}}(\hat{z})]$

Table 1: Posterior penalties utilized in the One-to-One and One-to-Many setting. These constraints softly enforce an alignment between control states and text spans by penalizing posterior violations. The objective sums over the three R_q in both cases.

generic semi-Markov algorithm (Sarawagi and Cohen, 2005). We store two tables β and β', both of size $T \times |\mathcal{C}|$. $\beta_t(c)$ denotes the event that there is a transition at time t from state c. $\beta'_t(c)$ denotes the event that there is a emission starting from time t at state c. Then we have the recursion for $\beta'_t(c)$ by "summing" over different span length, and we have the recursion for $\beta_t(c)$ that sums over all different state transitions.

The algorithm is generic in the sense that different $(\otimes, \oplus)$ operators allow us to compute different needed terms. For example, computing the partition function $Z = \sum_{z'} \phi(x, y, z')$ requires the $(+,\times)$ semiring (Goodman, 1999; Li and Eisner, 2009), other distributional terms can be computed by using the same algorithm with alternative semirings and backpropagation [3].

5 Posterior Constraints from Data Alignment

To make the PR model concrete, we consider the problem of incorporating weak supervision from heuristic alignment in a data-to-text generation task. Assume that we are tasked with describing a table x consisting of global field names $\mathcal{F}$ each with a text value v, e.g. $x_f = v$. Not all global fields may be used in a given x, we use $f \in x$ to indicate an active field.

x	name[Clowns] eatType[coffee shop], rating[1 out of 5], near[Clare Hall]
$f \in x$	name, eatType, rating, near
y	Clowns$_1$ is$_2$ a$_3$ coffee$_4$ shop$_5$ near$_6$ Clare$_7$ Hall$_8$ with$_9$ a$_{10}$ 1$_{11}$ out$_{12}$ of$_{13}$ 5$_{14}$ rating$_{15}$
$A(x, y)$	(1, 2, name), (4, 6, eatType), (7, 9, near), (11, 15, rating)

Table 2: Example of data alignment notation. Here x is a table of data, and f are its fields. For a given output y we enforce a soft alignment A.

We would like control states to indicate when each field is used in generation. Our alignment heuristic is that often these fields will be expressed using the identical text as in the table. While this heuristic obviously does not account for all cases, it is very common in natural language generation tasks as evidence by the wide use of copy attention based approaches (Gu et al., 2016; Gulcehre et al., 2016). To utilize these alignments, we use the notation $(i, j, f) \in A(x, y)$ to indicate that a span $i : j$ in the training text y overlaps directly with a field $f \in x$. Table 2 gives an example of the notation.

One-to-One Constraints We first consider one-to-one constraints where we assume that we have a static, mapping from fields to states $\sigma : \mathcal{F} \mapsto \mathcal{C}$. Given this mapping, we need to add penalties to encourage the semi-Markov model to overlap with the given weak supervision.

To enforce soft alignments, we define three posterior constraint types and their computation as shown in Table 1 (Left). The three constraints are i) Inclusion: if a span in y aligns with a field value

[3] We need four terms: (a) log-partition term $\log \sum_{z'} \phi(x, y, z')$ requires the log semiring $(\text{logsumexp}, +)$. The posterior marginals $q(z \mid x, y)$ requires backpropagating from the log-partition term; (b) max score $\max_z \phi(x, y, z)$: $(\max, +)$ max semiring and argmax $\arg\max_z \phi(x, y, z)$ by (subgradient) backpropagation, (c) entropy through an expectation semiring $\langle p_1, r_1 \rangle \otimes \langle p_2, r_2 \rangle = \langle p_1 p_2, p_1 r_2 + p_2 r_1 \rangle$, and $\langle p_1, r_1 \rangle \oplus \langle p_2, r_2 \rangle = \langle p_1 + p_2, r_1 + r_2 \rangle$, with $\mathbb{1} = \langle 1, 0 \rangle$. To initialize, all the emission, transition and length scores takes the form $\langle \phi, -\log \phi \rangle$. The algorithm returns $\langle Z, R \rangle$, and the true entropy is $\frac{R}{Z} + \log Z$. (d) exact sampling through one backward pass and one forward filtering backward sampling, where forward uses the log-partition semiring and backpropagation is by categorical sampling.

f, then label that span $\sigma(f)$ the state allocated to that field; ii) Exclusion: A span should only have a state $\sigma(f)$, if it aligns with the field value of type f; iii) Coverage. The usage count of state $\sigma(f)$ should be 1 if f in x.

One-to-Many Constraints We also consider the case when it is infeasible to specify a hard mapping σ between the fields and the states. For example, $\mathcal{F}$ could be unbounded or large, whereas we hope to keep the cardinality of states small for computational efficiency.

We propose a method of inducing a dynamic soft mapping $\sigma(c \mid f)$ as we train the model, and impose constraints on the mapping from table field to the state names. First, we would like the distribution of state given table field to be consistent, so one table field is mapped to roughly 1 state. Second, we want to make use of the state space as much as possible by requiring a diverse usage of states.

In order to enforce these properties we introduce the dynamic mapping as a second amortized variational distribution $\sigma(c \mid f; M) = \text{softmax}(Mf)$ which gives the probability that a table field f takes on state c. As shown in Table 1 (Right), we define three constraints that regularize the local q with respect to the global σ: i) Sparsity: Each vocabulary entry in σ should have low entropy; ii) Fit: The global σ should represent the class name distribution posterior of each table field by minimizing the cross entropy between types $\sigma(c \mid f)$ and tokens $q(z_{i:j} \mid x, y)$ for all $(i, j, f) \in A(x, y)$; iii) Diversity: the aggregate class label distribution over all the token in a sentence should have high entropy.

6 Related Work

In addition to previously mentioned work, other researchers have noted the lack of control of deep neural networks and proposed methods at sentence-level, word-level, and phrase-level. For example Peng et al. (2018) and Luo et al. (2019) control the sentiment in longer-form story generation. Others aim for sentence-level properties such as sentiment, style, tense, and specificity in generative neural models (Hu et al., 2017; Oraby et al., 2018; Zhang et al., 2018; Shen et al., 2017). Closest to this work is that of Wiseman et al. (2018) who control phrase-level content by using a neuralized hidden semi-Markov model for generation itself. Our work differs in that it makes no independence assumption on the decoder model, uses a faster training algorithm, and proposes a specific method for adding constraints. Finally, there is a line of work that manipulates the syntactic structure of generated texts, by using some labeled syntactic attribute (e.g., parses) or an exemplar (Deriu and Cieliebak, 2018; Colin and Gardent, 2018; Iyyer et al., 2018; Chen et al., 2019). While our work uses control states, there is no inherent assumption of compositional syntax or grammar.

Posterior regularization (PR) is mostly used in standard EM settings to impose constraints on the posterior distribution that would otherwise be intractable (or computationally hard) in the prior. Ganchev et al. (2010) applies posterior regularization to word alignment, dependency parsing, and part-of-speech tagging. Combining powerful deep neural networks with structured knowledge has been a popular area of study: Xu et al. (2019) applies PR to multi-object generation to limit object overlap; Bilen et al. (2014) focuses on object detection, and uses PR features to exploit mutual exclusion. In natural language processing; Hu et al. (2016a,b) propose an iterative distillation procedure that transfers logic rules into the weights of neural networks, as a regularization to improve accuracy and interpretability.

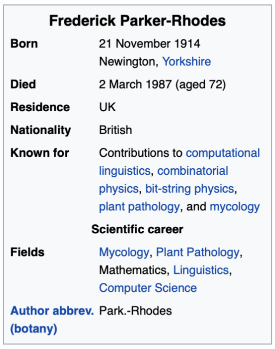
Finally, the core of this work is the use of amortized inference/variation autoencoder to approximate variational posterior (Kingma and Welling, 2014; Mnih and Gregor, 2014; Rezende et al., 2014). We rely heavily on a structure distribution, either linear chain or semi-Markov, which was introduced as a structured VAEs (Johnson et al., 2016; Krishnan et al., 2017; Ammar et al., 2014). Our setting and optimization are based on Kim et al. (2019), who introduce a latent tree variable in a variational autoencoding model with a CRF as the inference network, and on Yin et al. (2018) who use an encoder-decoder model as the inference network.

7 Experimental Setup

Data and Metrics We consider two standard neural generation benchmarks: E2E (Novikova et al., 2017) and WikiBio (Lebret et al., 2016a) datasets, with examples shown in Figure 1. The E2E dataset contains approximately 50K examples with 8 distinct fields and 945 distinct word types; it contains multiple test references for one source table. We evaluate in terms of BLEU (Papineni et al., 2002), NIST (Belz and Reiter, 2006), ROUGE-L

Table (x): name[Clowns] eatType[coffee shop] food[Chinese] customer-rating[1 out of 5] area[riverside] near[Clare Hall]

Ref.1: Clowns is a coffee shop in the riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .
Ref.2: The Chinese coffee shop by the riverside near Clare Hall that only has a customer rating of 1 out of 5 is called Clowns .
Ref.3: There is a Chinese coffee shop near Clare Hall in the riverside area called Clowns its not got a good rating though .



Frederick Parker-Rhodes

Born	21 November 1914 Newington, Yorkshire
Died	2 March 1987 (aged 72)
Residence	UK
Nationality	British
Known for	Contributions to computational linguistics, combinatorial physics, bit-string physics, plant pathology, and mycology

Scientific career

Fields	Mycology, Plant Pathology, Mathematics, Linguistics, Computer Science
Author abbrev. (botany)	Park.-Rhodes

Ref.1: Frederick Parker-Rhodes (21 March 1914 – 21 November 1987) was an English linguist, plant pathologist, computer scientist, mathematician, mystic, and mycologist.

Figure 2: Generation benchmarks. Model is given a table x consisting of semantic fields and is tasked with generating a description $y_{1:T}$ of this data. Two example datasets are shown. Left: E2E, Right: WikiBio.

(Lin, 2004), CIDEr (Vedantam et al., 2015) and METEOR (Lavie and Agarwal, 2007), using the official scoring scripts[4]. The WikiBio dataset contains approximately 700K examples, 6K distinct table field types, and 400K word types approximately; it contains one reference for one source table. We follow the metrics from (Lebret et al., 2016a) and evaluate the BLEU, NIST, and ROUGE-4 scores.

Architecture and Hyperparameters For all tasks, we use an encoder-decoder LSTM for the generative model. We follow recent state-of-the-art works in parametrizing our encoder, and we use copy attention and dual attention (Gu et al., 2016; Gulcehre et al., 2016; Liu et al., 2018): full model architectures are given in the supplement.

The inference network scores are computed using a BiLSTM. We compute the emission scores $\phi_{(e)}$ using span embeddings (Wang and Chang, 2016; Kitaev and Klein, 2018; Stern et al., 2017); transition scores $\phi_{(t)}$ by dot product between embedding vectors for the class labels; lengths $\phi_{(l)}$ is kept uniform, as in Wiseman et al. (2018). Additional details are in the supplement.

At training time, we use a rate for alleviating posterior collapse in the ELBO: warm-up the ELBO objective by linearly annealing the coefficient on the term $\sum_{t=1}^{T} \log p_\theta(z_t \mid z_{<t}, y_{<t})$ and $\mathrm{H}[q_\phi(z \mid x, y)]$ from 0 to 1, as implemented in Kim et al. (2019). We use the REINFORCE algorithm to do Monte Carlo estimation of the stochastic gradient. We choose the control variate to be the mean of the samples (Mnih and Rezende, 2016).

At decoding time, we only use the generative model. We use beam search with length normalization to jointly generate both the control states and the sentences. To obtain controlled generation, we observe the control states, and apply constrained beam search to $p(y \mid x, z)$.

Baselines For generation on E2E, we compare externally against 4 systems: E2E-BENCHMARK (Dušek and Jurčíček, 2016) is an encoder-decoder network followed by a reranker used as the shared task benchmark; NTEMP, a controllable neuralized hidden semi-Markov model; NTEMP+AR, the product of experts of both a NTemp model and an autoregressive LSTM network (Wiseman et al., 2018); SHEN19 (Shen et al., 2019) is an pragmatically informed model, which is the current state-of-the-art system on E2E dataset.

We also compare internally with ablations of our system: **ENCDEC** is a conditional model $p(y \mid x)$ trained without control states. **PC**^{0} is posterior control model with no constraints. It uses structured encoder with the PR coefficient set to 0. **PC**$^\infty$ is our model with hard constraints, which assumes fully-observed control states. These control states are obtained by mapping tokens with lexical overlap to their designated state; otherwise we map to a generic state. We train a seq2seq model $p(y, z \mid x)$ with full supervision of both control states and target text. Our main model is **PC**$^\lambda$, which applies PR with coefficient given by hyperparameter λ.

For WikiBio, we compare externally against 5 systems: NTEMP and NTEMP+AR as above; LEBRET16 (Lebret et al., 2016a), which uses copy attention and an NNLM; LIU18 (ENCDEC), which is our base encoder-decoder LSTM model, and LIU18 (Field Gating) which uses a field gating table encoder and a decoder with dual attention (Liu et al., 2018). For internal comparison on WikiBio, we compare between the one-to-one and one-to-

[4] Official E2E evaluation scripts available at `https://github.com/tuetschek/e2e-metrics`

			E2E		
	BLEU	NIST	ROUGE	CIDEr	MET
			validation		
E2E-Bench*	69.25	8.48	72.6	2.40	47.0
EncDec*	70.81	8.37	74.1	2.48	48.0
NTemp	64.53	7.66	68.6	1.82	42.5
NTemp+AR	67.70	7.98	69.5	2.29	43.1
PC^0	69.10	8.32	72.6	2.35	47.3
PC^∞	69.36	8.36	71.3	2.29	46.4
PC^λ	72.93	8.63	75.5	2.54	48.4
			test		
E2E-Bench*	65.93	8.59	68.5	2.23	44.8
Shen19*	68.60	8.73	70.8	2.37	45.3
EncDec*	66.34	8.55	68.0	2.18	44.3
NTemp	55.17	7.14	65.7	1.70	41.9
NTemp+AR	59.80	7.56	65.0	1.95	38.8
PC^λ	67.12	8.52	68.7	2.24	45.4

		WikiBio	
	BLEU	NIST	R-4
		test	
Lebret16*	34.7	7.98	25.8
Liu18(EncDec)*	43.7	-	40.3
Liu18(FieldGating)*	44.9	-	41.2
NTemp	34.2	7.94	35.9
NTemp+AR	34.8	7.59	38.6
$PC^\lambda_{\text{one-to-one}}$	44.7	9.92	43.3
$PC^\lambda_{\text{one-to-many}}$	44.2	9.59	41.5

Table 3: Automatic metrics for text generation. $*$ marks systems without learned control states. (Left) E2E. Comparison of systems from Dušek and Jurčíček (2016); Wiseman et al. (2018); Shen et al. (2019), our model and ablations. (Right) WikiBio. Comparison of Wiseman et al. (2018); Liu et al. (2018); Lebret et al. (2016a) and our full model.

many constraints in §5. $\mathbf{PC}^\lambda_{\textbf{one-to-one}}$ applies the One-to-One posterior constraints (left of Table 1). $\mathbf{PC}^\lambda_{\textbf{one-to-many}}$ applies the One-to-Many posterior constraints (right of Table 1).

8 Experiments

Table 3 shows the main results for the E2E and WikiBio, comparing to both standard neural models and controllable systems. On E2E (left), our posterior control model outperforms the neural benchmark system on all validation metrics and most of the test metrics. It also achieves results comparable or better than a specialized encoder-decoder system. It has significantly better performance than the controllable NTemp and NTemp+AR in all metrics on both validation and test. This demonstrates that the PC model provides interpretable and controllable states without sacrificing any representation power or generation performance.

For internal comparison, having soft constraints on the posterior outperforms the system PC^∞ (forced hard constraints) and PC^0 (no constraints). Anecdotally, we find that if two fields have the same value, then the hard coding system is often forced into the wrong decision. Similarly removing posterior regularization altogether leads to a slightly weaker performance than our controlled model.

On the larger WikiBio dataset (right) our model also significantly outperforms both the controllable NTemp and NTemp+AR baselines in all three metrics. It gives improvements over Liu et al. (2018)'s strong encoder-decoder style model. The promising result from WikiBio dataset suggests that the method scales to larger datasets and the PR style works well in handling large field spaces. In addition, we find that dynamic constraints are feasible compared with static constraints (we believe this is because the modeling burden on $PC^\lambda_{\text{one-to-many}}$ is heavier since it also needs to figure out the clustering). Overall, the dynamic framework opens up the possibility of generalizing to work well with a wider set of constraints.

9 Analysis

Qualitative Analysis Table 4 shows how control states (shown by different colors) are used in generated sentences. We use examples generated by the PC^λ system on the WikiBio dataset. We obtain outputs by beam search over control states and words. The first block contains examples with relatively complete coverage by the semantically grounded control states, including name, birth date, death date, occupation and nationality. We note that when a control state is selected, the textual span covered by the control state tend to respect truthfulness by copying from the table. The second block shows a longer example that uses less of the source, but still remain truthful with respect to the table.

Table 5 (left) qualitatively demonstrates the multi-modality of output of the system on E2E

PC^{λ}
billy ruge -lrb- c. 1885 – 1955 -rrb- was an american film actor .
debra dene barnes is an associate professor of piano studies at miss america 1968 .
shaalin zoya -lrb- born 22 february 1997 -rrb- is an indian actress .
carlos albert andrs -lrb- born february 24 , 1978 in madrid , spain -rrb- is a spanish sculptor .

Table (x): name[james horton]; birthdate[1850]; deathdate[none]; birthplace[boston, massachusetts]; allegiance[united states of America]; branch[united states navy]; rank[captain of the top]; awards[medal of honor]
REF: james horton -lrb- born 1850 -rrb- was a sailor serving in the united states navy who received the medal of honor for bravery .
PC^{λ}: james horton -lrb- born 1850 , date of death unknown -rrb- was a united states navy sailor and a recipient of the united states military 's highest decoration , the medal of honor .

Table 4: Qualitative examples on WikiBio dataset. (Top) Generated sentences control states highlighted. (Bottom) Full example of content selection with data table and reference. (Best viewed in color.)

dataset. We particularly note how the final system is trained to associate control states with field types. Here we fix the prior on z to 8 different sequences of class labels shown in different colors, and do constrained beam search on the generative model by holding z fixed, and decoding from the model $p_\theta(y \mid x, z)$.

Controllability Next we consider a quantitive experiment on model control. Assuming we have a mapping from control states to fields, ideally, at test time z should use the right states from the source x.[5] Let $\mathcal{S} = \{(i, j, f) : z_{i,j} = c, f \in x, \sigma(f) = c\}$ be the field states used by z. Define the field word overlap between x and y as,

$$\#match = \sum_{(i,j,f)\in\mathcal{S}} \text{unigram-overlap}(y_{i:j}, x_f)$$

We can compute *precision*, *recall*, and *coverage* under this metric,

$$\frac{\#match}{\sum_{(i,j,f)\in\mathcal{S}}(j-i)}, \quad \frac{\#match}{\sum_{f\in x}|x_f|}, \quad \frac{|\mathcal{S}|}{|c : c \in x|}.$$

Under these metrics we see the following control metrics on the E2E dataset,

	P	R	C
PC^{∞}	0.996	0.895	0.833
PC^{λ}	1.0	0.969	1.0

The PC model with soft posterior constraints performs better than having hard constraints on all three metrics. Having $P = 1$ means that the control states are a strong signal to copy from the table, and $C = 1$ means that control states learn to cover all table fields. On WikiBio, the model has a precision of 0.83 on the, meaning that on average, when we generate a good control state, 83% of the generated tokens will match the table content. Since only a fraction of the source table in WikiBio is used, recall and coverage are less applicable.

Distributional Metrics Table 5 (right) shows distributional metrics related to the optimization of the generative model and the inference network. The reconstruction perplexity, Rec. is much lower than the full perplexity, PPL and the KL divergence between the variational posterior and the conditional prior is highly non-zero. These observations indicate that latent variables are being used in a non-trivial way by the generative model. It also suggests the variational model is not experiencing posterior collapse.

Limitations Given the promise of PR as a technique for inducing control states, it is worth noting some of the current limitations to our specific application of the method. Currently, we use simple rules which do not generalize well to paraphrase. Our weak supervision relies on direct overlap to align states and fails on aligning phrases like `less then 10 dollars` that are expressed as `cheap`. Additionally, while at test time, our method is comparable to a standard decoder model, it does require slightly longer to train due to both the dynamic program and the requirement to compute multiple samples.

10 Conclusion

This work introduces a method for controlling the output of a blackbox neural decoder model to follow weak supervision. The methodology utilizes posterior regularization within a structured variational framework. We show that this approach can induce a fully autoregressive neural model that is as expressive as standard neural decoders but also utilizes meaningful discrete control states. We show this decoder is effective for text generation while inducing meaningful discrete representations.

[5] On E2E dataset, we remove the binary table field, "family friendly" which is never expressed by lexical match.

Table (x): name[Clowns] eatType[coffee shop] food[English] customerrating[5 out of 5] area[riverside] near[Clare Hall]
(1) Clowns is a 5 star coffee shop located near Clare Hall .
(2) Clowns is a coffee shop that serves English food and is near Clare Hall . It is in riverside and has a 5 out of 5 customer rating .
(3) Near Clare Hall in Riverside is coffee shop , Clowns . It serves English food , and has received a customer rating of 5 out of 5 .
(4) Near the riverside , Clare Hall is a coffee shop called Clowns that serves English food and has a customer rating of 5 - stars .
(5) Near Clare Hall , Clowns coffee shop has a five star rating and English food .
(6) Clare Hall is a 5 star coffee shop near to Clowns that serves British food .
(7) Clowns coffee shop is near Clare Hall in Riverside . It serves English food and has an excellent customer rating .
(8) 5 star rated restaurant , Clowns coffee shop is located near Clare Hall .

Models	Rec. ↓	PPL ↓	KL
	E2E		
PC^0	1.81	3.74	19.8
PC$^\lambda$	2.35	3.70	12.8
	WikiBio		
PC^0	2.57	3.82	10.69
PC$^\lambda_{\text{one-to-one}}$	2.45	4.07	10.19
PC$^\lambda_{\text{one-to-many}}$	2.59	4.58	13.07

Table 5: (Left) Example of controlled generation $p_\theta(y \mid x, z)$ on the source entity "Clowns" from E2E dataset. The color represents the class label of the token z. (Right) Metrics related to the generative model/inference network measured on both E2E and WikiBio. Rec. is reconstruction perplexity based on $\mathbb{E}_{q(z|x,y)}[\log p_\theta(y \mid, x, z)]$. PPL is the perplexity per token estimated by importance sampling.

Induction of grounded control states opens up many possible future directions for this work. These states can be used to provide integration with external rule-based systems such as hard constraints at inference time. They also can be used to provide tools for human-assisted generation. Another direction is to improve the sources of weak supervision and such as interactive new constraints provided by users. One could also explore alternative posterior constraints based on pre-trained models for summarization or paraphrase tasks to induce semantically grounded latent variables. Finally, it would be interesting to explore alternative training methods for these models, such as reducing reliance on hard sampling through better relaxations of structured models.

Acknowledgments

Thanks to Yoon Kim, Jambay Kinley, and Tristan Yang for ideas and discussion. AMR was supported by NSF CAREER 1845664 and IIS 1901030. XLL was supported by a Sony Research Award. We thank the anonymous reviewers for helpful comments.

References

Waleed Ammar, Chris Dyer, and Noah A. Smith. 2014. Conditional random field autoencoders for unsupervised structured prediction. *CoRR*, abs/1411.1147.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. 2014. Weakly supervised detection with posterior regularization. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984, Florence, Italy. Association for Computational Linguistics.

Emilie Colin and Claire Gardent. 2018. Generating syntactic paraphrases. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 937–943, Brussels, Belgium. Association for Computational Linguistics.

Jan Milan Deriu and Mark Cieliebak. 2018. Syntactic manipulation for generating more diverse and interesting texts. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 22–34, Tilburg University, The Netherlands. Association for Computational Linguistics.

Ondrej Dusek and Filip Jurcicek. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–51, Berlin, Germany. Association for Computational Linguistics.

M.J.F. Gales and Steve Young. 1993. The theory of segmental hidden markov models.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:20012049.

Joshua Goodman. 1999. Semiring parsing. *Comput. Linguist.*, 25(4):573–605.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany. Association for Computational Linguistics.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016a. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. 2016b. Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1670–1679, Austin, Texas. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. 2016. Composing graphical models with neural networks for structured representations and fast inference. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2946–2954. Curran Associates, Inc.

Yoon Kim, Alexander M. Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019. Unsupervised recurrent neural network grammars. *CoRR*, abs/1904.03746.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Rahul G. Krishnan, Uri Shalit, and David Sontag. 2017. Structured inference networks for nonlinear state space models. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 2101–2109. AAAI Press.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016a. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Rmi Lebret, David Grangier, and Michael Auli. 2016b. Neural text generation from structured data with application to the biography domain. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Zhifei Li and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 40–51, Singapore.

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

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning.

Fuli Luo, Damai Dai, Pengcheng Yang, Tianyu Liu, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Learning to control the fine-grained sentiment for story ending generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6020–6026, Florence, Italy. Association for Computational Linguistics.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Andriy Mnih and Karol Gregor. 2014. Neural variational inference and learning in belief networks. *CoRR*, abs/1402.0030.

Andriy Mnih and Danilo Rezende. 2016. Variational inference for monte carlo objectives. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2188–2196, New York, New York, USA. PMLR.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text 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)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. *CoRR*, abs/1706.09254.

Shereen Oraby, Lena Reed, Shubhangi Tandon, Sharath T.S., Stephanie Lukin, and Marilyn Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 180–190, Melbourne, Australia. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, New Orleans, Louisiana. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of ICML.*

Sunita Sarawagi and William W Cohen. 2005. Semi-markov conditional random fields for information extraction. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1185–1192. MIT Press.

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically informative text 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)*, pages 4060–4067, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6830–6841. Curran Associates, Inc.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575. IEEE Computer Society.

Wenhui Wang and Baobao Chang. 2016. Graph-based dependency parsing with bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315, Berlin, Germany. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, Brussels, Belgium. Association for Computational Linguistics.

Kun Xu, Chongxuan Li, Jun Zhu, and Bo Zhang. 2019. Multi-objects generation with amortized structural regularization. *arXiv preprint arXiv:1906.03923*.

Pengcheng Yin, Chunting Zhou, Junxian He, and Graham Neubig. 2018. Structvae: Tree-structured latent variable models for semi-supervised semantic parsing. *CoRR*, abs/1806.07832.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2018. Learning to control the specificity in neural response generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1108–1117, Melbourne, Australia. Association for Computational Linguistics.

Appendix

The generative model is an LSTM with two layers with hidden dimension equals 500, input dimension equals 400, and dropout of 0.2. The inference network uses a one-layer Bi-LSTM with hidden size of 500 and input size of 400 to encode the sentence. We use large max segment length, $L = 8$ (segmental for data-to-text) and $L = 1$ (linear chain for POS induction) and 0.2 dropout in the inference network. The Bi-LSTM used for encoding the source table is has hidden dimension of 300. Both the generative model and the inference network share word embeddings.

The batch size is 10 for WikiBio and 20 for PTB and E2E. The generative model and the inference network are optimized by Adam (Kingma and Ba, 2014) gradient clipping at 1, with learning rate of 0.002 and 0.001 respectively. Parameters are all initialized from a standard Gaussian distribution. The learning rate decays by a factor of two for any epoch without improvement of loss function on validation set, and this decay condition is not triggered until the eighth epoch for sufficient training. Training is done for max of 30 epochs and allows for early stopping.

For data-to-text problem, we need to encode the data table. We encode the E2E source table by directly concatenating word embeddings and field embeddings and indices for each token, for example, if the word w is the ith token from left and jth token from right under field type f, then we represent the token using a concatenation $[\text{emb}(w) \cdot \text{emb}(f) \cdot \text{emb}(i) \cdot \text{emb}(j)]$. We encode the WikiBio table by passing a bidirectional-LSTM through the tokens in the table, where each token has similar embedding by concatenation as above. The encoding of the table is denoted as c. We use copy attention (Gu et al., 2016; Gulcehre et al., 2016) in the generative model, and the attention vector α at a time step is parametrized by the class label z at that time step. Recall the contextual representation is $\sum_i \alpha_i \cdot c_i$, where $\alpha_i = \text{softmax}(\text{score}(h_t, c_i))$ and $\text{score}(h_t, c_i) = (W_z(h_t) + b_z) \cdot (W_2(c_i) + b_2)$, the parametrization from z happens during the feedforward network indexed by z. For the WikiBio data, we use a dual attention mechanism described in (Liu et al., 2018), where the first attention is the same as above and the second attention uses a different encoder context c_i', the c_i' only looks at the concatenation of field type and field index, but not the field value itself, i.e. $[\text{emb}(f) \cdot \text{emb}(i) \cdot \text{emb}(j)]$. Then the two attention forms two different sets of α_i and they are multiplied together and renormalized to form an attention.