POSTERIOR CONTROL OF BLACKBOX GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many tasks in natural language processing and related domains require high precision output that obeys dataset-specific constraints. This level of fine-grained control can be difficult to obtain in large-scale neural network models. In this work, we propose a structured latent-variable approach that adds discrete control states within a standard autoregressive neural paradigm. Under this formulation, we can include a range of rich, posterior constraints to enforce task-specific knowledge that is effectively trained into the neural model. This approach allows us to provide arbitrary grounding of internal model decisions, without sacrificing any representational power of neural models. Experiments consider applications of this approach for text generation and part-of-speech induction. For natural language generation, we find that this method improves over standard benchmarks, while also providing fine-grained control.

1 INTRODUCTION

A continuing challenge in the deployment of deep learning models for natural language processing is developing methods that ensure controlled outputs 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 other tasks like conditional natural language generation (NLG), where even the possibility of false outputs can make a system hard to use in realistic settings. While there have been significant improvements in generation quality from automatic systems (Mei et al., 2016; Dusek & Jurcicek, 2016; Lebret et al., 2016b), these methods are still far from being able to produce consistent output (Wiseman et al., 2017).

The dominant modeling paradigm in NLP is the neural encoder-decoder model, either built with RNNs or transformers (Vaswani et al., 2017). These models are unsurpassed in their ability to generate fluent output as well as produce useful representations of their source content. However, utilizing fully auto-regressive decoders prevents one from factoring out the concerns of a generation problem, as each part of the model is fully dependent. This issue makes it difficult to achieve outputs that follow their source conditioning while also incorporating domain constraints. Research into *controllable* deep models aims to circumvent the all-or-nothing dependency trade-off of encoder-decoder systems to allow for higher-precision systems.

There have been several proposals for controlling NLP models using deep generative models. One line of research has looked at higher-level control of trying to inject properties into standard deep decoders. For example, Hu et al. (2017) uses generative adversarial networks where the attributes of the text (e.g., sentiment, tense) are manipulated. Another alternative line of work has aimed at fine-grained properties but requires factoring the decoder to impose local constraints (Wiseman et al., 2018).

This work targets the benefits of both blackbox generation models and fine-grained control. We consider a fully autoregressive RNN model that can be used in a generic encoder-decoder system, but train it with structured latent variables to inject control states. This backbone makes it easy to incorporate external constraints at training time through posterior regularization to influence the model's decisions, while not requiring explicit factorization or test-time changes. These constraints allow us to ground the decisions of a neural model with explicit semantic information about the problem of interest, while not giving up any modeling power.

Technically, the approach utilizes recent advances in structured amortized variational inference to make training efficient and accurate. We also introduce an approach to amortized posterior

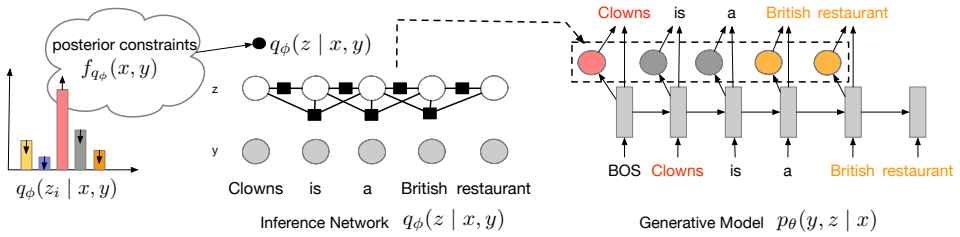

Figure 1: Model structure. (Middle) An inference network ϕ is used to parameterize a structured CRF $q_\theta(z \mid x, y)$. (Right) During training, samples from q are used to simulate the posterior control states of a blackbox generation model $p(y, z \mid x)$, shown by the colored circles. (Left) To ground the control states to represent problem-specific semantics, posterior regularization is used to enforce distributional constraints $f_q(x, y)$. The whole system is optimized end-to-end to learn latent properties (colors) of the final output tokens.

regularization to enforce additional constraints on the learned distribution. These constraints can be enforced explicitly through efficient structured expectation calculations. Overall the approach is fast to train, easy to deploy and can be added on to existing systems.

We demonstrate that the method can improve accuracy and control, utilizing a range of different posterior constraints, on several synthetic and real-world tasks, including text generation and part-of-speech inductions. In particular on two large-scale 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.

2 Control States in Blackbox Generation

We consider a generic sequence generation setting where the input consists of an arbitrary conditioning 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 $z_{1:T}$ each with a label in C. We assume that these states are weakly-supervised at training through problem-specific constraints. The goal is to induce a model of $p(y \mid x) = \sum_z p(y, z \mid x)$.

As a running example, we will consider a table-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 part of the table is being described, where our weak supervision corresponds to direct textual overlap.

Throughout, we will assume the generative model is a blackbox autoregressive (neural) decoder that produces both y and z. Define this model as:

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

For example, let $h_t(y_{1:t-1}, z_{1:t-1})$ be the hidden state at time-step t, e.g. of an RNN. We generate the latent class z_t and next token y_t by a softmax,

$$p_\theta(z_t \mid z_{<t}, y_{<t}) = \text{softmax}(W_0 h_t + b_0) \quad p_\theta(y_t \mid z_{\leq t}, y_{<t}) = \text{softmax}(W_1 [h_t, g_\theta(z_t)] + b_1)$$

where 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 term of interest will be the posterior distribution $p_\theta(z \mid x, y)$ which gives the probability over the control states. The model parameterization makes this distribution intractable to compute. To estimate this term, we use variational inference to define a parameterized variational posterior distribution, $q_\phi(z \mid x, y)$. This distribution is from a preselected family of possible distributions $\mathcal{Q}$.[1] To fit the parameters θ and variational parameters ϕ, we maximize a standard evidence lower bound.

$$\mathcal{L}(\theta) \geq \text{ELBO}(\theta, \phi) = \mathbb{E}_{z \sim q_\phi(z \mid x, y)}[\log p_\theta(y, z \mid x)] + \text{H}[q_\phi(z \mid x, y)] \tag{1}$$

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

While challenging to optimize, several 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 use this model as a backbone for enforcing problem-specific control by injecting weak supervision into the model p_θ through constraints on the variational posterior q_ϕ.

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., 2009). Traditionally this method used linear constraints with exponential family parameterizations. Within algorithms such as expectation maximization, this leads to convenient closed-form updates. However, this is infeasible with neural parameterizations. In this section, we develop alternative gradient-based optimizations for amortized variational inference.

Consider the maximum-likelihood objective, $\max_\theta \mathcal{L}(\theta)$. Posterior regularization modifies this objective based on distributional constraints on the posterior. Assume that we have a user-defined distributional property, $f_p(x, y)$, with target value b. The PR objective penalizes the maximum likelihood if the latent feature expectations diverge from this target.

$$\mathcal{L}_{PR}(\theta) = \mathcal{L}(\theta) - \min_{\phi,\xi} \mathrm{KL}[q_\phi(z \mid x, y) \mid\mid p_\theta(z \mid x, y)] + \lambda||\xi|| \text{ such that } f_{q_\phi}(x, y) - b \leq \xi$$

where to softly relax the target criteria, PR introduces two optimization terms: slack variables ξ, corresponding to how far the feature constraint is from being satisfied, and a surrogate posterior $q_\phi(z \mid x, y)$, to use for computing f. Alternatively we can consider a lower bound of the PR objective where we move the constraint to objective function.[2]

$$\mathcal{L}_{PR}(\theta) \geq \mathrm{PRLBO}(\theta, \phi) = \mathcal{L}(\theta) - \mathrm{KL}[q_\phi(z \mid x, y) \mid\mid p_\theta(z \mid x, y)] + \lambda||f_{q_\phi}(x, y) - b||$$

Problem-specific properties are encoded through the soft constraints, $f_{q_\phi}(x, y)$. For example, if we have partial information that the t'th control states takes on value k we can add a constraint $f_q(x, y) = q(z_t = k|x, y)$.[3] We might also consider other distributional properties, for instance the entropy of the marginal at position t, $f_q(x, y) = \mathrm{H}_{z'}(z_t = z'|x, y)$. Note that these constraints do not act on z directly but on the calculated posterior distribution. See §5 for more constraint examples.

Note that we can relate the q surrogate term in the PRLBO to the standard variational posterior in the ELBO simply by expanding the KL and rearranging terms.

$$\mathrm{PRLBO}(\theta, \phi) = \mathbb{E}_{q_\phi(z|x,y)} \log \frac{p_\theta(y, z \mid x)}{q_\phi(z \mid x, y)} + \lambda||f_{q_\phi}(x, y) - b|| = \mathrm{ELBO}(\theta, \phi) + \lambda||f_{q_\phi}(x, y) - b||$$

To train, we maximize both terms of the PRLBO, $\max_{\theta,\phi} \mathrm{PRLBO}(\theta, \phi)$ the model parameters θ and the variational parameters ϕ (which control both bounds). We use an amortized formulation for ϕ using a single inference network for all training examples. The key concern in computing PRLBO is being able to compute the ELBO (samples form q_ϕ and entropy) and the posterior constraint f_{q_ϕ}.

4 Inference with a Structured Variational Family

To efficiently calculate PRLBO, we pick a variational model class $\mathcal{Q}$ that form a structured (neural) conditional random field (CRF). Define for potentials f and factorization into parts $\mathcal{P}$,

$$q_\phi(z \mid x, y) = \frac{\phi(x, y, z)}{\sum_{z'} \phi(x, y, z')} \text{ where } \phi(x, y, z) = \prod_{p \in \mathcal{P}} \phi_p(x, y, z_p)$$

This formulation gives generic formulas for sampling, density calculation, entropy calculations, and calculation of marginal values $q(z_p \mid x, y)$. These marginals are useful for imposing posterior constraints in $f_q(x, y)$ on specific local decisions in the z. Furthermore for many classes of structured

[2] We need to consider the sign of the constraint, in particular, $f_{q_\phi}(x, y) - b \geq 0$ could guarantee the equivalence between the two formulations.

[3] This style of first-order constraints has a convenient optimization form in exponential-family expectation maximization.

CRFs used in NLP, these terms can be computed efficiently through generic, semiring dynamic programming.[4] Consider two examples:

Example 1: (Linear-Chain) Consider first a model utilizing local first-order dependencies on the z variables, and define our factorization to give potentials on these dependencies with first-order parts,

$$\phi(x, y, z) = \prod_{t=1}^{T} \phi_{t-1,t}(x, y, z_{t-1,t})$$

Under this model, all terms needed for PR and the PRLBO calculation can be computed efficiently using a forward dynamic program (similar to HMM). We note as well that the ϕ potentials can be parameterized by an arbitrary neural network over x, y. Constraints on $f_q(x, y) = q(z_t \mid x, y)$ can then be efficiently added.

Example 2: (Semi-Markov) A semi-Markov (or segmental) CRF (Gales & Young, 1993; Sarawagi & Cohen, 2005) is a richer sequence model that allows for spans of z to cover multiple tokens. Given a span i (inclusive) to j (exclusive) let $z_{i:j} = c$ indicate that the span has label c. We define our part set $\mathcal{P}$ to be possible consecutive spans $i : j$ and $j : k$. We restrict segments to a max length of L.

Algorithm 1: Semi-markov Algorithm

Given ϕ. Set $\beta_T(i) = \mathbb{1}\ \forall i \in \{1, 2, \cdots, K\}$;
for $t = T - 1$ *down to* 0 **do**
  for $i = 1$ *up to* K **do**
    $\beta_t^*(i) = \oplus_{d=1}^{\min(L,T-t)} \beta_{t+d}(i) \otimes \phi_{(l)}(d) \otimes \phi_{(e)}(x, y, i, t, t + d)$;
  for $i = 1$ *up to* K **do**
    $\beta_t(i) = \oplus_{j=1}^{K} \beta_t^*(j) \otimes \phi_{(t)}(i, j)$;
return $Z = \oplus_{i=1}^{K} \beta_0^*(i) \otimes \pi(i)$

Semi-Markov CRFs give potentials factorized into these neighboring spans, parametrized by emission scores, $\phi_{(e)}$; the transition scores, $\phi_{(t)}$; and length scores, $\phi_{(l)}$.

$$\phi(x, y, z) = \prod_{i,j,k \in \mathcal{P}} \phi_{(e)}(x, y, z_{i:j}, i, j) \cdot \phi_{(t)}(z_{i:j}, z_{j:k}) \cdot \phi_{(l)}(j - i) = \prod_{i,j,k \in \mathcal{P}} \phi_{i,j,k}(x, y, z_{i:j}, z_{j:k})$$

Marginals $q_\phi(z_{i:j} \mid x, y)$ represent the expected occurrence count for each labeled span, so we can easily penalize or reward each segment count to respect our prior knowledge. The generalized semiring computation (Sarawagi & Cohen, 2005) for these terms is given in Algorithm 1. [5]

Synthetic Experiment To demonstrate this appraoch, we generate data from a hidden semi-Markov model $p(y, z)$. Labels z are $C = \{1 \dots 5\}$ with $L = 5$ and y is tuples of the form "a-b", where first is the latent state and the second is a distractor. For example if z = "0, 0, 0, 4, 4, 3, 3, 4", and y ="0-1, 0-3, 0-5, 4-1, 4-4, 3-2, 3-3, 4-0". Segment length varies, but the total number of segments is constant at 4. A single PR constraint set f_q to be the expected segment length of z and $b = 4$.

Table 1 shows the results for a linear chain and semi-Markov CRF, whose max segment length is $L = 5$. The semi-Markov model achieves the best perplexity as it models the data. By itself, semi-Markov is not able to learn the correct segments, but with PR, it learns them nearly exactly (F_1). We also measure the reconstruction perplexity (Rec.), i.e., the perplexity given a posterior sample. Results for PR indicate that its control states are more useful for reconstruction (lower perplexity) than a model with no PR.

	SM: L=5		Chain	
PR	✓		✓	
PPL ↓	6.10	6.26	6.47	6.68
F_1 ↑	0.99	0.59	-	-
Rec. ↓	1.93	5.42	2.50	5.45

Table 1: Synthetic control experiment.

[4] Assume we have an algorithm for computing the partitional $Z = \sum_{z'} \phi(x, y, z')$ over the $(+,\times)$ semiring (Goodman, 1999; Li & Eisner, 2009). When this holds, other distributional terms can be computed by using the same algorithm with alternative semirings and backpropagation. These include (a) log-partition $\log \sum_{z'} \phi(x, y, z')$: (logsumexp, +) log semiring and marginals $q(z \mid x, y)$ by backpropagation; (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.

[5] The time complexity to compute the posterior moments of the full semi-Markov CRF is $O(|C|^2 nL)$. It is $O(|C|^2 n)$ for linear-chain CRF.

Source Table: 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

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

Figure 2: Table-to-text generation. 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.

5 Posterior Constraints for Natural Langauge Tasks

We now consider two very different example problems and propose a set of posterior constraints. The first table-to-text uses constraints that relate a sentence to its conditioning. The second part-of-speech induction uses a global parameter to enforce model-wide constraints.

Table-to-Text Assume that we are tasked with describing a table x consisting of fields $\mathcal{F}$ each with a text value. For example, a field might be of type "Restaurant Name" with field value "Tony's". We would like control states to indicate when each field is used. Our weak-supervision is that often these fields will be expressed using the same text as in the table. To enforce this, we will assume a predefined mapping $\sigma : \mathcal{F} \to C$ from table fields to class labels (which may or may not be unique). Our source of weak supervision will be when text in the generation overlaps directly with text in the data table. We use the notation $(i, j, c) \in F(x, y)$ to indicate a span in the training text y with class label $c = \sigma(f)$ overlaps directly with value in in x.

We define a set of three PR constraint types under a semi-Markov model to encode this weak supervision: i) if a span matches a field value f, then label that span $\sigma(f)$; ii) If a span has label $\sigma(f)$, then it should match a field value of type f; iii) The usage count of state $\sigma(f)$ should be 1 if f in x.

Name	Constraint
Inclusion	For all $(i, j, c) \in F(x, y)$, $q(z_{i:j} = c \mid x, y) \approx 1$
Exclusion	For all valid $(i, j, c) \notin F(x, y)$, $q(z_{i:j} = c \mid x, y) \approx 0$
Coverage	For all $f \in \mathcal{F}$ with $c = \sigma(f)$, $\sum_{(i,j)} q(z_{i:j} = c \mid x, y) - \mathbb{1}(f \in x) \approx 0$

Part-of-Speech Induction To demonstrate the versatility of this approach, we also consider a part-of-speech (POS) induction setting using a linear chain neural CRF with global constraints. We are given only a sentence $y_{1:T}$ with no conditioning. The weak supervision is that each word type should correspond to a sparse set of tags. For example, consider the word type "run". It can be used as a verb or a noun token, but it can never be used as an adjective, adverb or preposition, etc. In the unsupervised setting of POS induction, we regularize for sparsity of possible POS classes. Additionally, we want to avoid the degenerate case where all the word types are mapped to the same tag.

In order to enforce this constraint we introduce another amortized variational distribution $q'_M(c \mid w) = \text{softmax}(Mw)$ which gives the probability that word type w takes on tag c, i.e. a probabilistic tag dictionary. We define three constraints that regularize the local q with respect to the global q': i) Each vocabulary entry in q' should have low entropy; ii) The global q' should represent the POS distribution posterior of each word token by minimizing the cross entropy between types $q'(c \mid w)$ and tokens $q(z \mid y)$; iii) the aggregate POS distribution over all the token in a sentence should have high entropy.

Name	Constraint
Sparsity	For all $t \in 1 \dots T$ $\text{H}[q'(c \mid y_t)] \approx 0$
Fit	For all $t \in 1 \dots T$ $\text{H}[q'(c \mid y_t), q(z \mid y_t)] \approx \text{H}[q'(c \mid y_t)]$
Diversity	Let $\text{agg}(\hat{z}) \propto \sum_{t=1}^{T} q(z_t = \hat{z} \mid y)$ $\text{H}[\text{agg}(\hat{z})] \approx \text{H}[\text{Unif}(\hat{z})]$

6 Related Work

In addition to previously mentioned work, many other researchers have noted the lack of control of deep neural networks and proposed methods for controlled generation 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 & Cieliebak, 2018; Colin & 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 the non-neural structured prediction setting to impose constraints on the posterior distribution that would otherwise be intractable (or computationally hard) in the prior. Ganchev et al. (2009) 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 (VAE) to approximate variational posterior (Kingma & Welling, 2014; Mnih & 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 heavily 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 a seq2seq model as the inference network.

7 Data and Methods

We experiment with two different classes of tasks. Our main experiments are on conditional text generation in a table-to-text setting. We also consider unconditional induction of part-of-speech tags.

Data and Metrics For table-to-text, we use the 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 table fields and 945 distinct word types; it contains multiple references for one source table. We evaluate in terms of BLEU (Papineni et al., 2002), NIST (Belz & Reiter, 2006), ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015) and METEOR (Lavie & Agarwal, 2007), using the official scoring scripts[6]. 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.[7]

For part-of-speech (POS) induction, we use the Penn Treebank (Marcus et al., 1993), with train/valid/test splits from Dyer et al. (2016). Our preprocessing keeps the punctuation and a vocabulary size of 20K. Penn Treebank uses 36 distinct classes of POS tags for non-punctuation tokens and 8 classes for punctuation tokens. In preprocessing, we group the punctuation tokens to share one POS label "`PUNCT`." We use the gold POS tags from labeled Penn Treebank to evaluate parts-of-speech induction. The quality of part-of-speech induction is measured by V-measure (Rosenberg & Hirschberg, 2007). We also consider the perplexity of the decoder as well as its Reconstruction

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

[7]Scripts from file2rouge `https://github.com/pltrdy/files2rouge`

	E2E				
	BLEU	NIST	ROUGE	CIDEr	MET
			validation		
Benchmark	69.25	8.48	72.6	2.40	47.0
NTemp	64.53	7.66	68.6	1.82	42.5
NTemp+AR	67.70	7.98	69.5	2.29	43.1
RNN	70.63	8.09	71.9	2.21	45.7
Ours+Force	70.62	8.18	72.4	2.23	47.4
Ours−PR	71.45	8.18	73.0	2.30	48.0
Ours	73.58	8.31	76.2	2.43	50.1
			test		
Benchmark	65.93	8.59	68.5	2.23	44.8
NTemp	55.17	7.14	65.7	1.70	41.9
NTemp+AR	59.80	7.56	65.0	1.95	38.8
Ours	71.78	8.20	72.2	2.20	46.2

	WikiBio		
	BLEU	NIST	R-4
		test	
NTemp	34.2	7.94	35.9
NTemp+AR	34.8	7.59	38.6
NNLM	34.7	7.98	25.8
Liu et al. (2018)	44.9	-	41.2
Ours	46.1	9.21	41.4

Table 2: Automatic metrics for text generation. (Left) E2E. Comparison of systems from Dušek & Jurčíček (2016), Wiseman et al. (2018), our model and ablations. (Right) WikiBio. Comparison of Wiseman et al. (2018) Liu et al. (2018), Lebret et al. (2016a) and our full model.

perplexity, which is the perplexity under a sample from q_ϕ, indicating how well the model learns to use the control states.

Architecture and Hyperparameters For all tasks, we use an encoder-decoder LSTM for generation. We follow recent state-of-the-art work in designing our encoder and attention mechanisms (Gu et al., 2016; Gulcehre et al., 2016; Liu et al., 2018). Details are in the appendix. Automatic evaluation results from p are given using beam search by jointly generating the control states as well as the sentence.

The inference network is computed using a BiLSTM. We compute $\phi_{(e)}$ using the span method (Wang & Chang, 2016; Kitaev & Klein, 2018; Stern et al., 2017); $\phi_{(t)}$ by dot product between embedding vectors for the class labels. For $\phi_{(l)}$, we adapt the practice in Wiseman et al. (2018) to keep the length score uniform. Additional details are in the appendix. 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 a sample size of 5 for Monte Carlo estimation of the stochastic gradient and estimate the stochastic gradient by using the REINFORCE algorithm with a control variate computed as the mean of the samples (Mnih & Rezende, 2016).

Baselines For generation on E2E, we compare against several baselines: **Benchmark** (Dušek & Jurčíček, 2016), the task benchmark system of an encoder-decoder followed by a reranker; **RNN**: our encoder-decoder trained without latents; **Ours+Force**: our model with with hard constraints on the posterior instead of regularization. **Ours−PR** is an ablation study that drops PR from the full model. We compare with baselines from Wiseman et al. (2018): **NTemp**, a neuralized hidden semi-Markov model; **NTemp+AR**, the product of experts of both a NTemp model and an autoregressive LSTM network. Finally for WikiBio we compare against two encoder-decoder style models: **NNLM** (field & word) uses copy attention (Lebret et al., 2016a) and **Liu et al. (2018)** uses dual attention.

For the POS induction, our full model (**Ours**) is compared in perplexity against **RNNLM**, a standard LSTM language model with the same size as our model's autoregressive generative model. We do ablation studies on **Ours−PR**, the same model but drops PR; and **Ours+sup**, whose PR is supervised on gold POS tags.

8 Experiments

Generation Table 2 (left) shows the main results for E2E. On E2E, our model outperforms the benchmark system on all validation metrics and improves by 6 points of BLEU and 4 points of

Source: 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 .

Metric	Model	Valid	Test
PPL ↓	RNNLM	76.42	83.67
	Ours−PR	76.79	81.16
	Ours	76.59	80.83
	Ours+sup	82.55	85.87
Rec. ↓	Ours−PR	36.91	38.95
	Ours	13.36	14.07
	Ours+sup	21.54	22.33
V ↑	Ours−PR	0.245	0.249
	Ours	0.314	0.311
	Ours+sup	0.720	0.721

Table 3: (Left) Example of controlled generation $p_\theta(y \mid x, z)$ on the source entity "Clowns". The color represents the class label of the token z. (Right) POS induction results. Language modeling perplexity (PPL) upper bounds using Monte Carlo sampling, reconstruction perplexity by conditioning on a latent states (Rec.), and V-measurement (V).

ROUGE on test while being slightly worse in NIST and CIDEr. Similarly, it outperforms the controllable NTemp and NTemp+AR in all metrics on both validation and test. This demonstrates that in addition to providing constraints, PR can improve the accuracy of the model. We also consider alternatives approaches, including hard supervised training and training without PR. The empirical result suggests that forcing hard constraints does not preserve the generation performance as well as soft posterior regularization does. 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.

Table 2 (right) gives results for the larger WikiBio dataset. Again our model significantly outperforms both NTemp and NTemp+AR baselines in all three metrics. It also slightly outperforms Liu et al. (2018)'s 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.

Table 3 (left) qualitatively demonstrates output of the system. 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 $p_\theta(y \mid x, z)$.

Table 4 considers a quantitive experiment on model control. We define two metrics on how well the model aligns states with fields in x. Precision evaluates how well y's content matches the table content and recall evaluates how much table content is also mentioned in y. For example, $(i, j, c) \in z$ spans from i to j and labeled with c. We define a new operation for lookup, LU (c, x): this operation takes in a class label c and a table x, maps c to the corresponding field type in the table, and query the table to return the value of that field.

	P	R
Ours+Force	0.996	0.913
Ours	0.960	0.980

Table 4: Control metrics on E2E dataset.

$$\text{Precision}(x, y, z) = \underset{\substack{(i,j,c)\in z \\ \text{LU}(c,x)\neq\phi}}{\text{mean}} \frac{|y_{i:j} \cap c|}{j - i} \qquad \text{Recall}(x, y, z) = \underset{\substack{c \text{ such that} \\ \text{LU}(c,x)\neq\phi}}{\text{mean}} \frac{\sum_{(i,j,k)\in z: k=c} |y_{i:j} \cap \text{LU}(c, x)|}{|\text{LU}(c, x)|}$$

Our full model achieves a high level of control for both datasets. Hard coding ablation has a slightly better precision score but much lower recall. This is unsurprising by the design of hard-coding and its limitations.

Part-of-Speech Induction Table 3 (right) shows experiments on part-of-speech induction which is used to demonstrate the ability to include global posterior constraints. To begin, we find that our full model and Ours−PR are both comparable (slightly worse on validation and better on test) to RNNLM in perplexity. Adding direct supervision from the gold POS tags actually further harms LM perplexity, indicating that explicitly modeling the POS tags may actually hurt auto-regressive

language modeling in this setting. However, we do find that modeling tags changes other properties. Adding PR significantly reduces reconstruction perplexity — perplexity of $p(y \mid z)$, as shown by the Rec. part of the table. This indicates that the z learned from PR are more useful codes than the z model without PR. In fact, the reconstruction perplexity of PR is even lower than with supervision, indicating that the induced latent labels are more informative than standard POS tags.

The main result is that adding PR significantly improves the POS induction results on V-measure when compared with the base model without PR. We see that the global PR constraints can effectively move the model toward a better tag usage than the standard model. Ours+sup, being trained with supervision, scores much higher as an upper bound on the performance on this task.

Limitations Given the promise of PR as a technique for inducing control states, it is worth noting some of the current limitations to this method. Currently, our current approach does 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 longer to train due to both the dynamic program and the requirement to compute multiple samples. Both of these could potentially be addressed in future work.

9 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 an amortized structured variational framework. We show that this approach can induce a fully autoregressive neural model that is identical standard neural decoders but utilizes meaningful discrete control states. We show this decoder is effective for text generation and can also be used in induction settings such as unsupervised tagging. There are many possible future directions for this work. One direction is to improve the sources of weak supervision and make it trivial to specify new constraints. Another is to reduce the reliance on hard sampling through better relaxations of structured models. Finally, it would be interesting to try this approach for other blackbox neural models or develop general-purpose controlled modules.

References

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

Anja Belz and Ehud Reiter. Comparing automatic and human evaluation of NLG systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/E06-1040`.

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

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5972–5984, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1599. URL `https://www.aclweb.org/anthology/P19-1599`.

Emilie Colin and Claire Gardent. Generating syntactic paraphrases. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 937–943, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1113. URL `https://www.aclweb.org/anthology/D18-1113`.

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

Computational Linguistics. doi: 10.18653/v1/W18-6503. URL https://www.aclweb.org/anthology/W18-6503.

Ondrej Dusek and Filip Jurcicek. 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)*, 2016. doi: 10.18653/v1/p16-2008. URL http://dx.doi.org/10.18653/v1/P16-2008.

Ondřej Dušek and Filip Jurčíček. 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)*, pp. 45–51, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2008. URL https://www.aclweb.org/anthology/P16-2008.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 199–209, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1024. URL https://www.aclweb.org/anthology/N16-1024.

M.J.F. Gales and Steve Young. The theory of segmental hidden markov models. 01 1993.

Kuzman Ganchev, Joo Graa, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. 01 2009.

Joshua Goodman. Semiring parsing. *Comput. Linguist.*, 25(4):573–605, December 1999. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=973226.973230.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 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)*, pp. 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1154. URL https://www.aclweb.org/anthology/P16-1154.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 140–149, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1014. URL https://www.aclweb.org/anthology/P16-1014.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2410–2420, Berlin, Germany, August 2016a. Association for Computational Linguistics. doi: 10.18653/v1/P16-1228. URL https://www.aclweb.org/anthology/P16-1228.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1670–1679, Austin, Texas, November 2016b. Association for Computational Linguistics. doi: 10.18653/v1/D16-1173. URL https://www.aclweb.org/anthology/D16-1173.

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

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 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)*, pp. 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1170. URL https://www.aclweb.org/anthology/N18-1170.

Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. 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 (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2946–2954. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6379-composing-graphical-models-with-neural-networks-for-structured-representation pdf.

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

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. URL https://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2676–2686, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1249. URL https://www.aclweb.org/anthology/P18-1249.

Rahul G. Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 2101–2109. AAAI Press, 2017. URL http://dl.acm.org/citation.cfm?id=3298483.3298543.

Alon Lavie and Abhaya Agarwal. 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, pp. 228–231, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1626355.1626389.

Rémi Lebret, David Grangier, and Michael Auli. 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*, pp. 1203–1213, Austin, Texas, November 2016a. Association for Computational Linguistics. doi: 10.18653/v1/D16-1128. URL https://www.aclweb.org/anthology/D16-1128.

Rmi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016b. doi: 10.18653/v1/d16-1128. URL http://dx.doi.org/10.18653/v1/D16-1128.

Zhifei Li and Jason Eisner. 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)*, pp. 40–51, Singapore, August 2009. URL http://cs.jhu.edu/~jason/papers/#li-eisner-2009.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16599.

Fuli Luo, Damai Dai, Pengcheng Yang, Tianyu Liu, Baobao Chang, Zhifang Sui, and Xu Sun. Learning to control the fine-grained sentiment for story ending generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6020–6026, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1603. URL https://www.aclweb.org/anthology/P19-1603.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972470.972475.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 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*, 2016. doi: 10.18653/v1/n16-1086. URL http://dx.doi.org/10.18653/v1/N16-1086.

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

Andriy Mnih and Danilo Rezende. Variational inference for monte carlo objectives. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2188–2196, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/mnihb16.html.

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

Shereen Oraby, Lena Reed, Shubhangi Tandon, Sharath T.S., Stephanie Lukin, and Marilyn Walker. Controlling personality-based stylistic variation with neural natural language generators. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 180–190, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5019. URL https://www.aclweb.org/anthology/W18-5019.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pp. 43–49, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-1505. URL https://www.aclweb.org/anthology/W18-1505.

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

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D07-1043.

Sunita Sarawagi and William W Cohen. Semi-markov conditional random fields for information extraction. In L. K. Saul, Y. Weiss, and L. Bottou (eds.), *Advances in Neural Information Processing Systems 17*, pp. 1185–1192. MIT Press, 2005. URL http://papers.nips.cc/paper/2648-semi-markov-conditional-random-fields-for-information-extraction.pdf.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 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 (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6830–6841. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7259-style-transfer-from-non-parallel-text-by-cross-alignment.pdf.

Mitchell Stern, Jacob Andreas, and Dan Klein. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 818–827, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1076. URL https://www.aclweb.org/anthology/P17-1076.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pp. 4566–4575. IEEE Computer Society, 2015. ISBN 978-1-4673-6964-0. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2015.html#VedantamZP15.

Wenhui Wang and Baobao Chang. Graph-based dependency parsing with bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2306–2315, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1218. URL https://www.aclweb.org/anthology/P16-1218.

Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017. doi: 10.18653/v1/d17-1239. URL http://dx.doi.org/10.18653/v1/D17-1239.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. Learning neural templates for text generation. *CoRR*, abs/1808.10122, 2018. URL http://arxiv.org/abs/1808.10122.

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

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

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 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)*, pp. 1108–1117, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1102. URL https://www.aclweb.org/anthology/P18-1102.

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 & 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.