A causal framework for explaining the predictions of black-box sequence-to-sequence models: Supplementary Material

David Alvarez-Melis and Tommi S. Jaakkola
CSAIL, MIT
{davidam, tommi}@csail.mit.edu

A Formulation of graph partitioning with uncertainty

The bipartite version of the graph partitioning problem with edge uncertainty considered by Fan et al. (2012) has the following form. Assume we want to partition $U$ and $V$ into $K$ subsets each, say $\{U_i\}$ and $\{V_j\}$, with each $U_i$ having cardinality in $[c_{\text{min}}^i, c_{\text{max}}^i]$ and each $V_j$ in $[c_{\text{min}}^j, c_{\text{max}}^j]$. Let $x_{ik}^u$ be the binary indicator of $u_i \in U_k$, and analogously for $x_{jk}^v$ and $v_j$. In addition, let $y_{ij}$ be a binary variable which takes value 1 when $u_i, v_j$ are in different corresponding subsets (i.e. $u_i \in U_k, v_j \in V_{k'},$ and $k \neq k'$). We can express the constraints of the problem as:

$$
Y = \begin{align*}
\sum_{k=1}^{K} x_{ik}^u &= 1 & \forall i & (1) \\
\sum_{k=1}^{K} x_{jk}^v &= 1 & \forall j & (2) \\
c_{\text{min}}^i &\leq \sum_{i=1}^{N} x_{ik}^u \leq c_{\text{max}}^i & \forall i & (3) \\
c_{\text{min}}^j &\leq \sum_{j=1}^{N} x_{jk}^v \leq c_{\text{max}}^j & \forall j & (4) \\
- y_{ij} - x_{ik}^u + x_{jk}^v &\leq 0 & \forall i, j, k & (5) \\
- y_{ij} + x_{ik}^u - x_{jk}^v &\leq 0 & \forall i, j, k & (6) \\
x_{ik}^u, x_{jk}^v, y_{ij} &\in \{0, 1\}, & \forall i, j, k & (7)
\end{align*}
$$

Constraints (1) and (2) enforce the fact that each $s_i$ and $t_j$ can belong to only one subset, (3) and (4) limit the size of the $U_k$ and $B_k$ to the specified ranges. On the other hand, (5) and (6) encode the definition of $y_{ij}$: if $y_{ij} = 0$ then $x_{ik}^u = x_{jk}^v$ for every $k$. A deterministic version of the bipartite graph partitioning problem which ignores edge uncertainty can be formulated as:

$$
\min_{(x_{ik}^u,x_{ik}^y,y_{ij}) \in Y} N \sum_{i=1}^{M} \sum_{j=1}^{M} w_{ij}y_{ij} \tag{8}
$$

The robust version of this problem proposed by Fan et al. (2012) incorporates edge uncertainty by adding the following term to the objective:

$$
\max_{S: S \subseteq V \setminus S \leq |\Gamma|} \sum_{(i,j) \in J} \hat{a}_{ij} y_{ij} + (\Gamma - |\Gamma|) \hat{w}_{i,j} y_{i,j,t} \tag{9}
$$

where $\Gamma$ is a parameter in $[0, |V|]$ which adjusts the robustness of the partition against the conservatism of the solution. This term essentially computes the maximal variance of a single cut $(S, V \setminus S)$ of size $|\Gamma|$. Thus, larger values of this parameter put more on the edge variance, at the cost of a more complex optimization problem. As shown by Fan et al. (2012) the objective can be brought back to a linear form by dualizing the term (9), resulting in the following formulation

$$
\min \sum_{i=1}^{M} \sum_{j=1}^{M} w_{ij} y_{ij} + \Gamma p_0 + \sum_{(i,j) \in J} p_{ij} \tag{10}
$$

s.t. $p_0 + p_{ij} - \hat{a}_{ij} y_{ij} \geq 0, \ (i,j) \in J$
p_{ij} \geq 0, \ (i,j) \in J
p_0 \geq 0
(x_{ik}^u, x_{jk}^v, y_{ij}) \in Y,

This is a mixed integer programming (MIP) problem, which can be solved with specialized packages, such as GUROBI.

B Details on optimization and training

Solving the mixed integer programming problem (10) to optimality can be prohibitive for large
graphs. Since we are not interested in the exact value of the partition cost, we can settle for an approximate solution by relaxing the optimality gap tolerance. We observed that relaxing the absolute gap tolerance from the Gurobi default of $10^{-12}$ to $10^{-4}$ resulted in little minimal change in the solutions and a decrease in solve time of orders of magnitude. We added a run-time limit of 2 minutes for the optimization, though in all our experiments when never observed this limit being reached.

### C Details on the variational autoencoder

For all experiments in Sections 5.3 through 5.5 we use the same variational autoencoder: a network with three layer-GRU encoder and decoder and a stacked three layer variational autoencoder connecting the last hidden state of the encoder and the first hidden state of the decoder. We use a dimension 500 for the hidden states of the GRUs and 400 for the latent states $z$. We train it on a 10M sentence subset of the English side of the WMT14 translation task, with KLD and variance annealing, as described in the main text. We train for one full epoch with no KLD penalty and no noise term (i.e. decoding directly from the mean vector $\mu$), and start variance annealing on the second epoch and KLD annealing on the 8th epoch. We train for 50 epochs, freezing the KLD annealing when the validation set perplexity deteriorates by more than a pre-specified threshold.

Once trained, the variational autoencoder is used as a subroutine of SOC\textsc{R}AT to generate perturbations as described in Algorithm 2. Given an input sentence $x$, we use the encoder to obtain approximate posterior parameters $(\mu, \sigma)$, and then repeatedly sample latent representations from the a gaussian distribution with these parameters. The scaling parameter $\alpha$ constrains the locality of the space from which examples are drawn, by scaling the variance of the encoded representation’s approximate posterior distribution. Larger values of $\alpha$ encourage samples to deviate further away from the mean encoding of the input $\mu$, and thus more likely to result in diverse samples, at the cost of potentially less semantic coherence with the original input $x$. In Table 1 we show example sentences generated by this perturbation model on two input sentences from the WMT14 dataset with increasing scaling parameter $\alpha$.

### Algorithm 1 Variational autoencoder perturbation model for sequence-to-sequence prediction

1: procedure \textsc{Perturb}(x)
2: \hspace{1em} $(\mu, \sigma) \leftarrow \text{Encode}(x)$
3: \hspace{1em} for $i = 1$ to $N$ do
4: \hspace{2em} $\tilde{z}_i \sim N(\mu, \text{diag}(\alpha \sigma))$
5: \hspace{2em} $\tilde{x}_i \leftarrow \text{Decode}(\tilde{z}_i)$
6: \hspace{1em} end for
7: \hspace{1em} return $(\{\tilde{x}_i\}_{i=1}^N)$
8: end procedure

### D Black-box system specifications

The three systems used in the machine translation task in Section 5.3 are described below.

#### Azure’s MT Service

Via REST API calls to Microsoft’s Translator Text service provided as part of Azure’s cloud services.
Neural MT System  A sequence-to-sequence model with attention trained with the Open-NMT library (Klein et al., 2017) on the WMT15 English-German translation task dataset. A pre-trained model was obtained from http://www.opennmt.net/Models/. It has two layers, hidden state dimension 500 and was trained for 13 epochs.

A human  A native German speaker, fluent in English, was given the perturbed English sentences and asked to translate them to German in one go. No additional instructions or context were provided, except that in cases where the source sentence is not directly translatable as is, it should be translated word-to-word to the extent possible. The human’s German and English language models were trained for 28 and 16 years, respectively.

References
