Learning from Multiple Sources for Data-to-Text and Text-to-Data

Song Duong, Alberto Lumbreras, Mike Gartrell, Patrick Gallinari

▶ To cite this version:

HAL Id: hal-04002656

https://hal.sorbonne-universite.fr/hal-04002656v1

Submitted on 23 Feb 2023

Learning from Multiple Sources for Data-to-Text and Text-to-Data

Song Duong[1,2]   **Alberto Lumbreras**[1]   **Mike Gartrell**[1]   **Patrick Gallinari**[1,2]

[1]Criteo AI Lab
[2]Sorbonne Université, CNRS, ISIR, Paris, France

Abstract

Data-to-text (D2T) and text-to-data (T2D) are dual tasks that convert structured data, such as graphs or tables into fluent text, and vice versa. These tasks are usually handled separately and use corpora extracted from a single source. Current systems leverage pre-trained language models fine-tuned on D2T or T2D tasks. This approach has two main limitations: first, a separate system has to be tuned for each task and source; second, learning is limited by the scarcity of available corpora. This paper considers a more general scenario where data are available from multiple heterogeneous sources. Each source, with its specific data format and semantic domain, provides a non-parallel corpus of text and structured data. We introduce a variational auto-encoder model with disentangled style and content variables that allows us to represent the diversity that stems from multiple sources of text and data. Our model is designed to handle the tasks of D2T and T2D jointly. We evaluate our model on several datasets, and show that by learning from multiple sources, our model closes the performance gap with its supervised single-source counterpart and outperforms it in some cases. [1]

1 INTRODUCTION

Data-to-text (D2T) (Kukich, 1983; McKeown, 1985) is the task of converting structured data into fluent text. This generic task is involved in several applications: generating

---

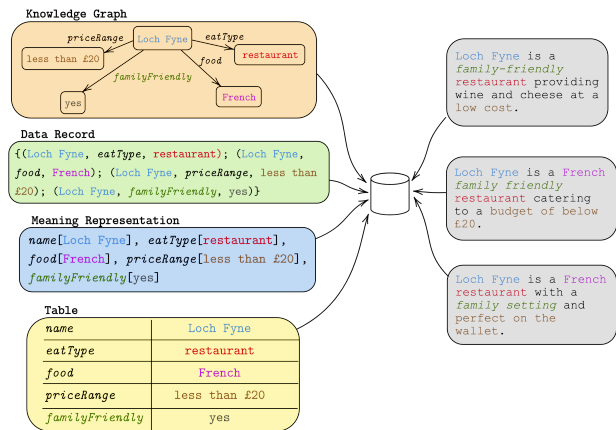[1]The code is available at github.com/sngdng/MSUnsupVAE.

Figure 1: The same content can be represented by different structures and by different texts. We show here an example from the E2E dataset (Dušek et al., 2019). On the left side, from top to bottom: knowledge graph, data record, meaning representation, and table.

summaries from patient records (Scott et al., 2013) or from tables such as summaries of sports games (Wiseman et al., 2017), generating product descriptions from databases in e-commerce (Wang et al., 2017), and so on. What makes this task different from conventional text generation is that structured data comes in many forms, e.g. web tables, databases, knowledge graphs, and so D2T systems have to handle these heterogeneous formats. Text vocabulary can also vary greatly from one domain to another, and systems need to adapt to this diversity as well (see Fig. 1). In addition, both modalities, text and data, are often noisy and incomplete, and in most practical situations there is no aligned corpus available, or at best small hand-crafted corpora. As a concrete example, consider the generation of product descriptions on an e-commerce website. Items can be described either as structured tables extracted from databases or as text. The former provides details of the characteristics of the items, while the latter summarises this information for the user. Information can be obtained from different providers and in different formats, also known as different sources (Ng et al., 2000). Even from the same

vendor, descriptions can vary widely depending on the product category (e.g. high-tech products and clothing). They may be quite noisy, such as unstructured sentences or series of keywords instead of a plain description, and in many cases only the text or data is available. In this case, the supervised formulation falls short, as we need to be able to use non-parallel data.

Text-to-data (T2D) is the dual task that aims at extracting relevant information from text and identifying their relations in order to generate structured data in a pre-defined format. Current T2D approaches are mainly limited to the conversion from text to graphs (Ye et al., 2021; Agarwal et al., 2020; Paolini et al., 2021). Until now, both tasks have been handled separately through different methods.

Recent advancements in LLMs have enabled encoding various forms of information in text and viewing D2T and T2D as dual tasks within a unified framework. However, existing systems have primarily focused on simplified scenarios to meet these requirements:

(i) First, to the best of our knowledge, all models to date consider a single source for data and text, and deal with a data format and a text style as defined by the training corpus. However, in practice, one is faced with heterogeneous formats, where data is collected from multiple sources, each with its own specificity. This also holds for related NLP tasks including *Question Answering* (Chen et al., 2020b; Talmor and Berant, 2019; Nan et al., 2022) and *Formal-Language-to-Text* (Shu et al., 2021; Chen et al., 2020c). This variety of data sources has led to the development of systems adapted to each case, resulting in architectures designed specifically for knowledge graphs (Ke et al., 2021; Zhao et al., 2020), tables (Liu et al., 2017; Rebuffel et al., 2020; Cao, 2020; Liu et al., 2019), and so on. Recent work unifies multiple sources of structured data while considering a diversity of tasks, such as *Structured Knowledge Grounding* (SKG) (Xie et al., 2022), by grounding both user requests and structured data in a common structured template.

(ii) Secondly, most models are trained in a supervised setting that requires the manual construction of aligned datasets. The latter are usually limited in size; for instance, one of the main D2T reference datasets, WebNLG (Gardent et al., 2017b), is composed of only 35,000 aligned pairs of text and knowledge graphs. Some recent works (Jin et al., 2020; Lebret et al., 2016; Chen et al., 2021a) have proposed larger datasets, e.g., by collecting texts from Wikipedia and extracting knowledge graphs from DBPedia (Lehmann et al., 2015) or Wikidata (Vrandečić and Krötzsch, 2014). However, as these text-graph pairs have been generated automatically, they are usually loosely aligned or simply non-parallel. Some works (Schmitt et al., 2020; Guo et al., 2020b, 2021) have tackled the non-parallel setting in the presence of a single source of knowledge graphs.

This paper considers the dual tasks of D2T and T2D within a unified formalism for multi-source conversion. We assume that structured data and texts are collected from multiple sources, with each source in a specific structured format. Our objective is to develop a unique system capable of performing conversion in both directions for a specified number of sources, where we convert data from a given source to text and, given a text, we convert it to any format from the set of targeted sources. With parameters shared between sources and tasks, the model can take full advantage of all sources, as opposed to systems trained on each source independently. We make the additional assumption that for training, data are collected from each source independently and are not aligned, i.e., there is no unique correspondence between a text and data. Given the difficulty of generating parallel datasets for D2T and T2D, this is a realistic scenario that allows us to leverage larger corpora in many practical situations. Note that even for aligned corpora, most of them are generated automatically, so the correspondence can be extremely loose. Compared to the usual single-source supervised or unsupervised settings, where a one-to-one mapping is assumed between text and data, we consider here a new problem corresponding to a many-to-many mapping between structured data and text: data comes from multiple heterogeneous sources, while text can express the same content in multiple ways.

Our contributions are the following:

• We frame D2T and T2D as two complementary sequence-to-sequence tasks that can be alternately optimized in an end-to-end, cycle-training fashion via iterative back-translation.

• We propose a latent variable model for handling multi-source D2T and T2D through a shared model common to both tasks and to all the sources. Our approach is based on a pre-trained T5 language model (Raffel et al., 2020) and leverages the variational autoencoder (VAE) formulation.

• We evaluate our method on several D2T datasets. We show that by learning from multiple sources, our model closes the performance gap with its supervised single-source counterparts on many datasets and even outperforms them on the DART dataset (Nan et al., 2021). This demonstrates the utility of learning from multiple heterogeneous sources in a unified way. Although, for practical reasons, our experiments have been conducted with the "T5-small" model, the results remain competitive with current state-of-the-art obtained with larger models on a portion of the datasets.

2 RELATED WORK

Data-to-text (T2D). Recent neural approaches have trained end-to-end models with either encoder/decoder architectures (Wiseman et al., 2017; Gardent et al., 2017a),

or using neural networks as part of a pipeline system (Moryossef et al., 2019; Castro Ferreira et al., 2019; Puduppully et al., 2019; Laha et al., 2020). In particular, using large pre-trained language models has been shown to greatly improve performance, either by simply framing data-to-text as a text-to-text task (Kale and Rastogi, 2020), or by using a GNN-based planner for graph-to-text tasks (Ribeiro et al., 2021; Zhao et al., 2020; Guo et al., 2020a). Current works mainly consider single-source problems. Some works have considered leveraging multiple sources, either by aggregating multiple structured datasets into a single-format dataset (Chen et al., 2020a; Nan et al., 2021), or by grounding user requests and structured data in a common structured template (Xie et al., 2022).

Text-to-data (T2D). Converting unstructured text to structured data can be framed as a specific instance of (open) information extraction tasks, such as Named Entity Recognition (NER) (Nadeau and Sekine, 2007; Putthividhya and Hu, 2011) or, more commonly, joint entity and relation extraction. Generally, given a text, the T2D task requires identifying the relevant entities and the relations between them, with the result being encoded as a triplet (Entity1, Relation, Entity2). Recent approaches consider the graph as a sequence of triples and cast the problem as a text-to-text task. They typically combine an encoder with an autoregressive decoder (Ye et al., 2021; Agarwal et al., 2020; Paolini et al., 2021). There has been less work on converting text to tables, although some recent approaches aim to generate tables from queries (Gupta and Berberich, 2019) in the information retrieval domain.

Cycle training with back-translation for unsupervised learning. Back-translation (Cheng et al., 2016; He et al., 2016; Cotterell and Kreutzer, 2018) has been shown to be very effective in unsupervised machine translation (Artetxe et al., 2018; Lample et al., 2018). Given non-parallel samples (x, y) from a source and target language, this approach jointly learns two models (from one language to another and vice-versa) by leveraging cycle-consistencies $x \rightarrow \hat{y} \rightarrow x$ and $y \rightarrow \hat{x} \rightarrow y$. Lample et al. (2018) identified a fundamental principle for successful backtranslation, namely that each model must be trained on a specific "language modeling objective to generate correct sentences in its language", i.e., each model must be able to capture the specifics of its own source domain.

Because of the duality of D2T and T2D, some works have framed both tasks as translation tasks between languages by linearizing the structured data. Following Lample et al. (2018), Schmitt et al. (2020) introduced a denoising objective for graph-to-text, and apply iterative back-translation to train a single sequence-to-sequence model (LSTM with attention and a copy mechanism) to perform both graphto-text and text-to-graph. Guo et al. (2020b) also leveraged back-translation to perform unsupervised learning, but considered separate graph-to-text and text-to-graph compo

nents. The former component is based on a pre-trained language model and takes a linearized graph as input, while the latter component uses an off-the-shelf entity extraction model and simply learns a relation classification model to obtain the graph. This breaks the symmetry of the two tasks, and it seems that the denoising objectives of Schmitt et al. (2020) are no longer needed for backtranslation to succeed. A few recent works have incorporated cycle-consistent approaches for the semi-supervised setting. Domingo et al. (2022) propose training a single T5 model for both graph-to-text and text-to-graph tasks, while Chang et al. (2021) proposes various data augmentation approaches and uses cycle consistency to ensure that the augmented data samples can be correctly reconstructed after having been formulated as text, and vice versa.

Variational framework for data-to-text. The variational autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014) is a generative model capable of leveraging latent variables. Bowman et al. (2016) was the first to train a text VAE using an RNN encoder/decoder model, with KL annealing used to mitigate posterior collapse.

In the D2T literature, variational models are used to induce diversity or to enable more fine-grained control over text generation from data. Guo et al. (2021) proposed a latent variable model to represent the diversity in text, allowing the model to learn a one-to-many mapping from the graph to the text domain. This approach performs both graph-to-text and text-to-graph tasks and uses the same standard NER component as in Guo et al. (2020b). Ye et al. (2020) proposed a variational model with disentangled content and style latent variables to perform the graphto-text task. Their semi-supervised algorithm is also inspired by back-translation but still requires some parallel data. For the table-to-text task, Wiseman et al. (2018) modeled templates as a sequence of discrete latent variables and learns them using hidden semi-Markov models. Puduppully et al. (2022) proposed a variational approach to model latent plans in order to perform content planning in D2T generation. Chen et al. (2021b) introduced a VAE to model the causality of logical table-to-text as a latent variable.

3 PROBLEM DEFINITION

Multi-source scenario. We consider the scenario where we have multiple data sources at our disposal, and the data from each source are collected independently. Each source consists of a set of texts $\{x_n\}_{n=1}^{N}$ and a set of single-format structured data $\{y_m\}_{m=1}^{M}$. The formats of the different sources may be heterogeneous, e.g., tables of different types, graphs, etc. We consider structured data formats that can be represented or mapped to a set of relational triples (subject, relation, object), as is the case for graphs, data records, and tables (see Fig. 1).

D2T & T2D tasks. Our goal is to solve the D2T and T2D

tasks. For the D2T task, we are given structured data from a source, and we seek to generate fluent text that explains the data. Conversely, for the T2D task, we are given a text and a target format indicator, and we aim to generate the corresponding structured data that summarize the text. We make the assumption that for D2T, multiple sentences can describe the same data, since there are many ways to verbalize a given piece of content. For T2D, given a target format, we assume that the mapping is unique since the content is uniquely encoded in the text.

Non-parallel setting. We study a general setting where each source consists of non-parallel sets of text and data, respectively: $(X_N, Y_M) = \left(\{x_n\}_{n=1}^N, \{y_m\}_{m=1}^M\right)$. In other words, during training, we assume that the origin of the text and data is known, but their alignment is unknown. Our goal is to train a single system that can perform the conversion in both directions for all sources. We denote $p_{\text{src}}(x, y)$ as the unknown joint distribution of the text and data from a source. In a non-parallel setting, since we do not know the correspondence, the samples in the dataset (X_N, Y_M) are assumed to come from the marginals, which are respectively p_T and p_D for the text and the data:

$$p_T(x) = \int_y p_{\text{src}}(x, y) dy, \quad (1)$$

$$p_D(y) = \int_x p_{\text{src}}(x, y) dx. \quad (2)$$

We assume that text and data are not statistically independent, i.e., $p_{\text{src}}(x, y) \neq p_T(x) p_D(y)$, so that it is possible to learn mappings $p_T(x|y)$ and $p_D(y|x)$ between the text and the data.

4 BACKGROUND: ITERATIVE BACK-TRANSLATION

Before delving into our proposed model in Section 5, in this section we formalize iterative back-translation, which is a fundamental part of our model derivation, and formally link our approach to the classical back-translation loss objective (Eq. 7). In the discussion that follows, we consider for simplicity a single-source objective. This is not limiting, since the multi-source problem amounts to solving several single-source problems with a shared model.

In the non-parallel setting, we do not have access to ground-truth parallel pairs $(x, y) \sim p_{\text{src}}(x, y)$, which makes the learning of $p_T(x|y)$ and $p_D(y|x)$ difficult since we cannot directly maximize the conditional likelihoods. We address this issue by introducing latent variables $y_n^{(t)}$ and $x_m^{(d)}$, corresponding respectively to the unobserved ground-truth pair or *missing pair* for our observed text $x_n^{(t)}$ and data $y_m^{(d)}$, as shown in Fig 2. Our goal is to learn the densities $p_T(x|y)$ and $p_D(y|x)$, which we parameterize respectively with $\theta^{(t)}$ and $\theta^{(d)}$. We denote by $\theta = [\theta^{(t)}, \theta^{(d)}]$
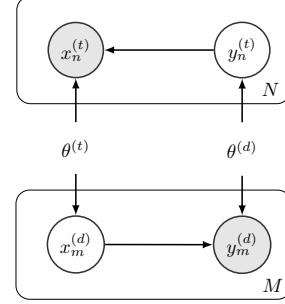

Figure 2: Graphical representation of the unsupervised setting where aligned pairs are not available.

the set of parameters of our model. Given any arbitrary observed text $x_n^{(t)}$, we can express its log-likelihood using the missing data distribution $p_{\theta^{(d)}}(y_n^{(t)})$ (see Appendix A.1) as

$$\log p_\theta(x_n^{(t)}) = \mathbb{E}_{p_{\theta^{(d)}}(y_n^{(t)})} [\log p_{\theta^{(t)}}(x_n^{(t)}|y_n^{(t)})]$$
$$+ D_{KL}(p_{\theta^{(d)}}(y_n^{(t)}) \| p_{\theta^{(d)}}(y_n^{(t)}|x_n^{(t)})). \quad (3)$$

Likewise, the log-likelihood of any given data $y_m^{(d)}$ can be obtained the same way using $p_{\theta^{(t)}}(x_m^{(d)})$:

$$\log p_\theta(y_m^{(d)}) = \mathbb{E}_{p_{\theta^{(t)}}(x_m^{(d)})} [\log p_{\theta^{(d)}}(y_m^{(d)}|x_m^{(d)})]$$
$$+ D_{KL}(p_{\theta^{(t)}}(x_m^{(d)}) \| p_{\theta^{(t)}}(x_m^{(d)}|y_m^{(d)})). \quad (4)$$

Given each equation, we do not know the distribution of the missing data $p_{\theta^{(d)}}(y_n^{(t)})$ nor of the missing text $p_{\theta^{(t)}}(x_m^{(d)})$. An arbitrary choice of these distributions would lead to an arbitrary estimation of the mapping functions $p_{\theta^{(t)}}(x_n^{(t)}|y_n^{(t)})$ and $p_{\theta^{(d)}}(y_m^{(d)}|x_m^{(d)})$. Instead, we follow the iterative back-translation scheme proposed in Hoang et al. (2018) and consider that

$$p_{\theta^{(d)}}(y_n^{(t)}) = p_{\theta^{(d)}}(y_n^{(t)}|x_n^{(t)}), \quad (5)$$
$$p_{\theta^{(t)}}(x_m^{(d)}) = p_{\theta^{(t)}}(x_m^{(d)}|y_m^{(d)}). \quad (6)$$

From a probabilistic perspective, iterative back-translation can be seen as successively updating the posterior of the missing data (resp. text) using the conditional likelihood of the observed data (resp. data) from the other model component.

Note that under the iterative back-translation scheme, optimizing the log-likelihoods is equivalent to optimizing the cycle objective for unsupervised learning (see Appendix A.2) from prior work (Hoang et al., 2018; Guo et al., 2020b, 2021):

$$\mathcal{L}_{\text{cycle}}(\theta^{(t)}, \theta^{(d)}) = \mathbb{E}_{p_T(x)} \left[\mathbb{E}_{p_{\theta^{(d)}}(y|x)} [\log p_{\theta^{(t)}}(x|y)] \right]$$
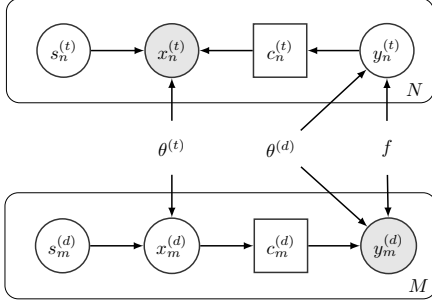$$+ \mathbb{E}_{p_D(y)} \left[\mathbb{E}_{p_{\theta^{(t)}}(x|y)} [\log p_{\theta^{(d)}}(y|x)] \right]. \quad (7)$$

Figure 3: Graphical representation of our model. We augment the existing variables in Fig. 2 with additional latent variables $s_n^{(t)}$ and $s_m^{(d)}$ to capture the diversity in the text, and condition the generation of the structured data on the data format.

5 AUGMENTED LATENT VARIABLE MODEL FOR DATA-TO-TEXT AND TEXT-TO-DATA

Our model, shown in Fig. 3 and described below, is a refined version of the general scheme illustrated in Fig. 2. Because of the non-alignment assumption, the corresponding model leads to intractable expressions of the likelihood. Therefore, we use a variational autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014) to make learning and inference tractable.

5.1 Augmented Latent Variable Model

Probabilistic graphical model. In Section 3, we assumed that the content from structured data can be expressed in multiple ways, leading to different textual surface realizations. This is modeled by conditioning the text generation on a latent variable denoted s for *style*. We make use of two such variables, $s_n^{(t)}$ and $s_m^{(d)}$, respectively, for conditioning on observed text occurrences $x_n^{(t)}$ and latent texts $x_m^{(d)}$, as shown in Fig. 3. These style variables will be useful for modeling the diversity of texts - thus potentially allowing us to generate diverse texts from a single piece of data at inference time, and for modeling the uncertainty of the association between text and data during training. Similarly, we introduce a parameter, denoted f, that specifies the targeted conversion format for a text. Since the target format is available at both test and training time, this is a fixed parameter and not a latent one.

ELBO. We choose to leverage the VAE framework for our model. We show that the cycle-training objective can be deduced from the graphical model in a way that is similar to an iteration of the Wake-Sleep algorithm (see Appendix A.3) (Cotterell and Kreutzer, 2018). Therefore, we derive the evidence lower bound (ELBO) equations for the text x

and the structured data y as:

$$\mathcal{L}_x(\theta^{(t)}, \phi) = \mathbb{E}_{p_{\theta^{(d)}}(\hat{y}|x;f)} \left[\mathbb{E}_{q_\phi(s|x)} \left[\log p_{\theta^{(t)}}(x|\mathrm{Enc}_\phi(\hat{y}), s) \right] \right]$$
$$- D_{KL}(q_\phi(s|x)\|p(s)), \qquad (8)$$

$$\mathcal{L}_y(\theta^{(d)}, \phi) = \mathbb{E}_{p(s)} \left[\mathbb{E}_{p_{\theta^{(t)}}(\hat{x}|y,s)} \left[\log p_{\theta^{(d)}}(y|\mathrm{Enc}_\phi(\hat{x}); f) \right] \right], \qquad (9)$$

where Enc_ϕ is the sequence-to-sequence encoder used for learning representations of both text and data (see Section Content encoding).

Priors. We place a Gaussian prior over the style of the text,

$$p(s_n^{(t)}) = p(s_m^{(d)}) = \mathcal{N}(0, I).$$

Since we want to incorporate information from observed text and data into their corresponding latent variables, we follow the iterative back-translation scheme described in Section 4, Eqs. 5 and 6, and define the priors as

$$p_{\theta^{(d)}}(x_m^{(d)}) = p_{\theta^{(d)}}(x_m^{(d)}|y_m^{(d)}), \qquad (10)$$
$$p_{\theta^{(t)}}(y_n^{(t)}) = p_{\theta^{(t)}}(y_n^{(t)}|x_n^{(t)}). \qquad (11)$$

Variational posteriors. For the variational posterior of the style, we use different parameterizations for $s_n^{(t)}$ and $s_m^{(d)}$, since in the former case we have access to observed text $x_n^{(t)}$, while for the latter there is no style information in the observed $y_m^{(d)}$. We parameterize $q_\phi(s_n^{(t)}|x_n^{(t)})$ and $q_\phi(s_m^{(d)}|y_m^{(d)})$ with a family of Gaussian distributions:

$$q_\phi(s_n^{(t)}|x_n^{(t)}) = \mathcal{N}(\mu_\phi(x_n^{(t)}); \Sigma_\phi(x_n^{(t)})), \qquad (12)$$
$$q_\phi(s_m^{(d)}|y_m^{(d)}) = \mathcal{N}(0, I). \qquad (13)$$

Content encoding. We initially considered encoding the content stochastically using a family of Gaussian distributions as in Ye et al. (2020). However, we found in preliminary experiments that considering the content as stochastic leads to a performance decrease, especially when regularizing with a Gaussian prior. Therefore, we follow a simpler approach and choose to encode the content $c_n^{(t)}$ and $c_m^{(t)}$ deterministically by setting

$$c_n^{(t)} = \mathrm{Enc}_\phi(y_n^{(t)}); \ c_m^{(d)} = \mathrm{Enc}_\phi(x_m^{(d)}), \qquad (14)$$

where Enc_ϕ is the encoder of the T5 model in our case.

Text-to-data inference. In order to infer data y from text x, we first pass a one-hot encoded data format vector f through a linear layer $s_f = W_f f + b_f$, and then feed it along with the encoded content $c = Enc_\phi(x)$ to the decoder to generate $y \sim p_\theta(y|c, s_f)$.

Data-to-text inference. Here we clarify how to infer text from data, i.e., $p_\theta(x|y)$. During training, the text style latent code is encoded from the text distribution, which we
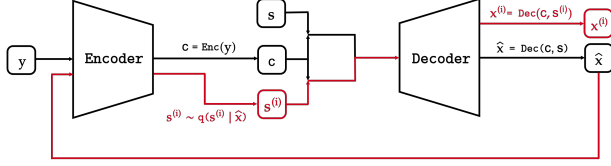
Figure 4: Given an observed data y, we generate a synthetic text $\hat{x}$ (black path). We use the synthetic text to retrieve the corresponding style posterior $q(s|\hat{x})$. We can now sample multiple style latent codes $s^{(i)} \sim q(s^{(i)}|\hat{x})$ and feed them into the decoder, along with the encoded content $c = \text{Enc}(y)$, to generate texts with multiple styles $x^{(i)} = \text{Dec}(c, s^{(i)})$ (red path).

do not have at inference time. The usual way to infer text from data in this scenario is to sample from the prior, as the divergence term in the ELBO pushes the variational posterior to its respective prior. Empirically, we found that this sampling method is acceptable when the dimensionality of the latent code is low. However, in the case of a high-dimensional latent space, the model fails to learn. We suspect that this is due to the inherent discrete nature of text; the latent space of the VAE for text exhibits regions of low density and can contain multiple holes (Xu et al., 2020; Li et al., 2021), leading to the generation of incoherent text when naively decoding a latent code sampled from the prior. This effect is stronger in our case, since we need to perform inference starting at the very beginning of training. This results in noisy and uninformative samples, and leads to the failure of iterative back-translation.

In order to stabilize the inference process, when optimizing for structured data y, instead of sampling from the prior, we choose to fix the style latent code during training to $s = [0, ..., 0]$ (the mean of the Gaussian prior) in order to infer synthetic text via $\hat{x} \sim p_\theta(x|c, s)$, where $c = Enc_\phi(y)$. At inference time, to mitigate the train-test discrepancy, we infer $p_\theta(x|y)$ by first generating $\hat{x} \sim p_\theta(x|c, s)$ with $s = [0, ..., 0]$ and $c = \text{Enc}_\phi(y)$, and then use this synthetic sample to generate the style latent code $s \sim q_\phi(s|\hat{x})$. We can then infer $x \sim p_\theta(x|c, s)$ (see Fig. 4).

Disentanglement. As we encode the style of the text and the content using the same encoder, it is likely that both variables are entangled. We introduce a way to control their degree of entanglement, since this may influence the performance of the model. One way to assess the level of disentanglement between two random variables X and Y is by computing their Mutual Information (MI), and disentangling X from Y is achieved by minimizing their MI. While some works have explicitly sought to minimize this quantity or an estimate of it (Sanchez et al., 2019; Cheng et al., 2020a,b; Colombo et al., 2021), other works (Wang et al., 2018; Chang et al., 2022) have simply chosen to introduce a bottleneck on the capacity of the style encoder, constraining the dimensionality of one of the two variables

in order to limit the information flow from one variable to the other, also known as the *content leakage* phenomenon (Hu et al., 2020). We follow the latter approach and show (see Appendix. A.4) that we can upper-bound the MI between the content and the style variables with a bound that scales with the dimension of the style of the text and the structured data format variables.

5.2 Training

Algorithm 1: Cycle-training via back-translation

Input: $\mathbf{X}^{(t)}$, $\mathbf{Y}^{(d)}$, format f, Decoder θ, Encoder ϕ
while *not converged* **do**

$\quad (x, y) \sim \left(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)} \right)$
$\quad$ `// Denoising objectives`
$\quad \nabla(\theta, \phi) \leftarrow \nabla_{(\theta,\phi)} \mathcal{L}_{denoising}(\theta, \phi) \quad$ (Eq. 15)
$\quad$ `// T2D` $(y \rightarrow \hat{x} \rightarrow y)$
$\quad s \sim \mathcal{N}(0, I)$ (or $s \leftarrow [0, ..., 0]$)
$\quad \hat{x} \sim p_\theta(x \mid y, s) \qquad$ `/* no gradient */`
$\quad c \leftarrow Enc_\phi(\hat{x})$
$\quad \nabla(\theta, \phi) \leftarrow \nabla(\theta, \phi) + \nabla_{(\theta,\phi)} \mathcal{L}_y(\theta, \phi) \quad$ (Eq. 9)
$\quad$ `// D2T` $(x \rightarrow \hat{y} \rightarrow x)$
$\quad \hat{y} \sim p_\theta(y \mid x, f) \qquad$ `/* no gradient */`
$\quad c \leftarrow Enc_\phi(\hat{y})$
$\quad \nabla(\theta, \phi) \leftarrow \nabla(\theta, \phi) + \nabla_{(\theta,\phi)} \mathcal{L}_x(\theta, \phi) \quad$ (Eq. 8)
$\quad$ `// Update`
$\quad (\theta, \phi) \leftarrow (\theta, \phi) + \nabla(\theta, \phi)$
end

One potential danger that can arise when only optimizing the cycle objective is a collapse onto the trivial identity mapping, which may happen when the source and target domains are the same. We encounter this phenomenon due to our linearization of the structured data, where we implicitly frame the D2T and T2D tasks as translation tasks. Learning the trivial identity mapping for both tasks is one way to locally maximize the objective. To remedy this issue, Lample et al. (2018) and Schmitt et al. (2020) introduce denoising objectives, which make the model robust to noise. We follow the same methodology as Schmitt et al. (2020) and adapt their noise functions (*swap*, *drop*, *blank*, *repeat*, *rule*) for all structured data formats (see Appendix A.5 for examples).

In addition to the ELBO, we also optimize the following denoising objective:

$$\mathcal{L}_{denoising}(\theta) = \underset{x \sim p_T(x), \tilde{x} \sim Noise(x)}{\mathbb{E}} \left[\log p_\theta(x|\tilde{x}) \right]$$
$$+ \underset{y \sim p_D(y), \tilde{y} \sim Noise(y)}{\mathbb{E}} \left[\log p_\theta(y|\tilde{y}) \right]. \quad (15)$$

Iterative back-translation leverages the use of synthetic samples. When optimizing for a text x, we sample synthetic data $y \sim p_{\theta^{(d)}}(y|x)$, with $\theta^{(d)}$ being fixed, and we do

the same with $\theta^{(t)}$ when optimizing for a structured data y. Since we are jointly optimizing Eq. 8 and Eq. 9 using the same model $\theta = (\theta_t, \theta_d)$, we accumulate gradients for text and data before updating the parameters of the model. Our method is described in Algorithm 1.

6 EXPERIMENTS

6.1 Datasets

We simulate a heterogeneous multi-source setting by considering a collection of D2T datasets. Each dataset uses a specific structured encoding. Since they are originally designed for a supervised setting, they are all aligned. We simply remove this alignment information and consider each source as a set of text and structured data. All the datasets are available from the GEM Benchmark (Gehrmann et al., 2021). Due to the heterogeneous sizes of these datasets, we use a temperature mixing of $T = 2$, as in Raffel et al. (2020), to train our models in the multi-source setting.

WebNLG 2020 (English) (Castro Ferreira et al., 2020) is a dataset composed of 35,426 pairs of knowledge graphs and text crawled from 15 categories of DBpedia. This is follow-up work from the WebNLG 2017 challenge (Gardent et al., 2017b).

Cleaned E2E (Dušek et al., 2019) results from cleaning the original E2E NLG dataset (Novikova et al., 2017), an English benchmark dataset for data-to-text models that verbalizes a set of 2–9 key-value attribute pairs in the restaurant domain. The cleaned version has filtered out examples with hallucinations and outputs that do not fully cover all input attributes.

DART (Nan et al., 2021) is an English dataset composed of 62,659 parallel text-data pairs, which aggregates multiple data-to-text datasets, including E2E and WebNLG, into a common format called data records. These data records consist of sets of triples and can be seen as a flat representation of the graph structure.

ToTTo (Parikh et al., 2020) is a high-quality English table-to-text dataset with more than 120,000 examples, where each example is composed of a table from Wikipedia with highlighted cells that is paired with a sentence that describes these cells. All examples in this dataset were post-edited in multiple steps to ensure that the targets are fully faithful to the input information.

6.2 Baselines

We consider the following baselines for our experiments (MS holds for multi-source, SS for single-source):

MS-Sup: This pre-trained T5 model is fine-tuned on all datasets in a multi-source, *supervised* setting. We train a

separate model for D2T and T2D. For T2D, we consider one prefix for each format and train in a multi-task fashion.

SS-Sup: This pre-trained T5 model is fine-tuned on each dataset independently and is *supervised*. We train a separate model for D2T and T2D.

MS-UnSupVAE: Our approach.

SS-UnSupVAE: This model is a single pre-trained T5 model fine-tuned on each dataset separately for both D2T and T2D. This approach is close to the one in Guo et al. (2021), but they use separate models for D2T and T2D and additional preprocessing for T2D.

MS-UnSup: This pre-trained T5 model is fine-tuned on all datasets in a multi-source, *unsupervised* setting. We consider a prefix for each format and train a single model for both D2T and T2D in a multitask fashion.

SS-UnSup: This model uses a pre-trained T5 model on each dataset separately for both D2T and T2D. This is comparable to Schmitt et al. (2020), which uses a BiLSTM-LSTM architecture as the encoder-decoder instead of a T5 model in our case.

Note that because of limited resources, we conducted all of our experiments using the *T5-small* model. However, we will see that for some experiments we reach the performance of state-of-the-art *T5-large* models.

6.3 Evaluation metrics

D2T metrics. We assess traditional metrics in our experiments, including BLEU (Papineni et al., 2002; Post, 2018), ROUGE-L (Lin, 2004), and METEOR (Banerjee and Lavie, 2005; Denkowski and Lavie, 2014).

T2D metrics. Little has been established in the literature for the evaluation of generated structured data. We follow the same approach as prior works on text-to-graph (Guo et al., 2020b, 2021; Schmitt et al., 2020) and report the precision, recall, and f1 score of predicted entities and relations for all formats. In addition to these metrics, we also compute the SemBLEU score (Song and Gildea, 2019) by converting all formats into a graph structure. This score, which is adapted to the structure of graphs, provides more information on higher-order n-grams than previous entity and relation scores.

Diversity metrics. Since our model has the ability to generate diverse texts from the same data, it is important to assess the effectiveness of this mechanism. We evaluate the diversity of the generated texts in our experiments using the Self-BLEU metric (Zhu et al., 2018), which treats one generated sentence as the hypothesis and the others as the reference, and calculates the BLEU score for every set of generated sentences. The score is then averaged over the test set. We also report Distinct-1 and Distinct-2 (Li

Table 1: Evaluation results for D2T and T2D

	DART		WEBNLG		E2E		TOTTO	
	BLEU	**METEOR**	**BLEU**	**METEOR**	**BLEU**	**METEOR**	**BLEU**	**METEOR**
MS-Sup	**52.613**	**38.29**	**52.037**	**35.85**	40.679	35.3	**44.196**	**37.32**
SS-Sup	45.163	35.91	50.502	35.58	**40.95**	**35.43**	43.74	37.08
MS-UnSupVAE (ours)	**51.074**	**37.47**	**49.575**	**34.3**	39.33	34.94	**36.793**	**36.11**
SS-UnSupVAE	39.23	33.35	43.73	32.5	39.31	35.02	32.85	33.83
MS-UnSup	48.435	36.2	45.172	31.72	**39.529**	**34.95**	35.129	34.92
SS-Unsup	39.27	33.19	44.72	32.59	39.36	34.79	33.40	34.21

	DART			WEBNLG			E2E			TOTTO		
	ENT F1	**REL F1**	**SemBLEU**	**ENT F1**	**REL F1**	**SemBLEU**	**ENT F1**	**REL F1**	**SemBLEU**	**ENT F1**	**REL F1**	**SemBLEU**
MS-Sup	**92.01**	**81.36**	**69.1**	**73.59**	**47.5**	**47.55**	**97.87**	**97.64**	**99.21**	**49.82**	**15.04**	11.9
SS-Sup	84.78	71.49	54.5	70.84	45.46	39.87	96.97	96.61	98.85	49.62	14.03	**12.82**
MS-UnSupVAE (ours)	84.64	72.65	67.73	65.38	39.22	35.62	88.16	88	95.32	47.02	13.76	15.14
SS-UnSupVAE	72.56	56.03	39.71	57.22	25.37	16.09	85.57	85.18	93.04	43.36	10.21	5.72
MS-UnSup	80.74	67.69	56.6	60.45	33.73	26.97	86.44	86.13	95.08	44.95	11.92	10.19
SS-Unsup	71.29	55.12	42.96	57.3	25.47	13.62	78.96	78.59	80.56	44.26	10.52	8.06

et al., 2016), which corresponds to the number of distinct unigrams and bigrams in generated texts. This metric is then scaled by the total number of generated tokens, which penalizes long sentences.

6.4 Results

We present some of our results in Table 1. A more complete list of results can be found in Table 2 (Appendix).

D2T. (i) Supervised vs Unsupervised. The supervised model **MS-Sup**, which is effectively an upper bound on model performance, performs the best as expected. (ii) Single source vs multi-source. Except for E2E, multi-source models always outperform their single-source counterparts. Learning from multiple sources improves performance across datasets. Since the E2E dataset is a very specific, small, and closed domain (restaurant) dataset, training on multiple domains may introduce interference. (iii) VAE vs non VAE. For the multi-source setting, the VAE version **MS-UnSupVAE** always outperforms its non-VAE counterpart **MS-UnSup** by a significant margin. This result highlights the importance of uncertainty modeling when the conversion problem becomes more challenging, as is the case for multi-source vs single-source scenarios. (iv) Unsupervised models. Except on E2E, where all models have similar performance, our model outperforms all other unsupervised approaches across all metrics, and closes the performance gap with the supervised baselines. Note that **MS-UnSupVAE** is particularly effective on DART, outperforming even the single-source supervised baselines and current state-of-the-art obtained with larger models (see Appendix 2). We attribute this performance to the multi-source nature of DART. By flattening knowledge graphs into sets of triples, the data records in DART have lost information that was present in the original data. Overall, it appears that training on heterogeneous formats that originate from the same content improves model performance.
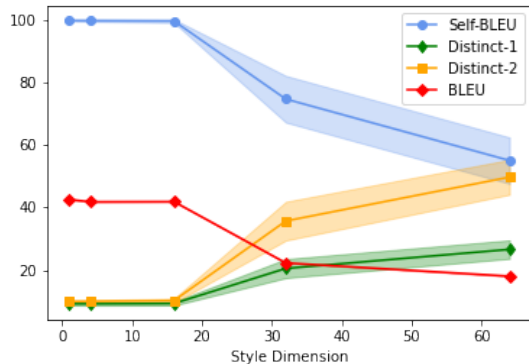

Figure 5: Evolution of average BLEU, Distinct-1, Distinct-2 (higher is better), and self-BLEU (lower is better) metrics over all datasets as the dimensionality of the style increases.

T2D. First, we notice that using a text-to-text model for T2D results in some format errors, although they are not frequent (less than 5% for all datasets). Training on multiple sources is challenging for the T2D task, as it also increases format errors due to the consideration of several formats, such as for the **MS-Sup** model on the E2E dataset. Compared to their supervised counterparts, all unsupervised approaches underperform on entity and relation extraction. Nevertheless, our model remains competitive with the other supervised baselines.

Diversity Results. We jointly evaluate the diversity and quality of the text generated by our model as the dimension of the style latent variable varies. Due to the stochasticity of text generation, we report the mean and 95% confidence intervals for the diversity metrics and the BLEU metrics (quality) for 10 different random seeds. We average the metrics over all datasets. We observe that increasing the dimensionality of the style vector increases the Distinct-(1,2) scores, i.e., the number of unigrams and bigrams increases.

The generated sentences are also more diverse, since Self-BLEU decreases. It appears that there is little variation in the generation when the style dimensionality is fairly low (less than 16). As the dimensionality increases, however, the model tends to hallucinate, as the BLEU metrics generally decrease (see more examples in Appendix A.7).

7 ASSUMPTIONS AND LIMITATIONS

One limitation of our work is that we have considered sources with structured data formats that can be mapped to sets of relational triples. The treatment of more diverse structured formats and tasks remains an open problem. A second limitation concerns the diversity in the text domain and in the relation types considered in different corpora. Although our model behaves well on sources with different text domains and relations, the types of relations remain relatively simple, typical of the benchmarks in this domain, and extension to more complex situations is challenging. A potentially limiting assumption concerns our non-parallel framework for D2T and T2D. Although we remove alignment from the data by breaking aligned pairs, the domains of the two modalities remain aligned, which differs from the non-parallel framework considered in machine translation (Lample et al., 2018), where no alignment is assumed. This is due to the lack of large open-domain D2T datasets.

8 CONCLUSION

We have proposed a model capable of learning D2T and T2D mappings in a non-parallel setting, where text and data come from multiple sources. Our experiments show that the model is able to leverage this diversity to provide improved performance compared to unsupervised baselines. Our results suggest that mixing multiple formats can potentially improve model performance in both the parallel and non-parallel setting. The advantage of our method lies in its applicability to more realistic problems, where the data is heterogeneous, or where a strict match between text and data cannot be found. A promising avenue for future work would be to consider the adaptation to scenarios where new sources with scarce data are made available, requiring few-shot learning capability.

Acknowledgements

References

O. Agarwal, M. Kale, H. Ge, S. Shakeri, and R. Al-Rfou. Machine translation aided bilingual data-to-text generation and semantic parsing. In *WebNLG+*, pages 125–130. ACL, 2020.

M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *ICLR*, 2018.

S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72. ACL, 2005.

S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21. ACL, 2016.

J. Cao. Generating natural language descriptions from tables. *IEEE Access*, 8:46206–46216, 2020. doi: 10.1109/ACCESS.2020.2979115.

T. Castro Ferreira, C. van der Lee, E. van Miltenburg, and E. Krahmer. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *EMNLP-IJCNLP*, pages 552–562. ACL, 2019.

T. Castro Ferreira, C. Gardent, N. Ilinykh, C. van der Lee, S. Mille, D. Moussallem, and A. Shimorina. The 2020 bilingual, bi-directional webnlg+ shared task overview and evaluation results (webnlg+ 2020). In *WebNLG*, pages 55–76. Association for Computational Linguistics, 2020.

E. Chang, X. Shen, D. Zhu, V. Demberg, and H. Su. Neural data-to-text generation with LM-based text augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 758–768. ACL, 2021.

J.-H. R. Chang, A. Shrivastava, H. Koppula, X. Zhang, and O. Tuzel. Style equalization: Unsupervised learning of controllable generative sequence models. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 2917–2937. PMLR, 2022.

M. Chen, S. Wiseman, and K. Gimpel. WikiTableT: A large-scale data-to-text dataset for generating Wikipedia article sections. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 193–209. ACL, 2021a.

W. Chen, Y. Su, X. Yan, and W. Y. Wang. KGPT: Knowledge-grounded pre-training for data-to-text generation. In *EMNLP*, pages 8635–8648. ACL, 2020a.

W. Chen, H. Zha, Z. Chen, W. Xiong, H. Wang, and W. Y. Wang. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036. ACL, 2020b.

W. Chen, J. Tian, Y. Li, H. He, and Y. Jin. De-confounded variational encoder-decoder for logical table-to-text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5532–5542. ACL, 2021b.

Z. Chen, W. Chen, H. Zha, X. Zhou, Y. Zhang, S. Sundaresan, and W. Y. Wang. Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111. ACL, 2020c.

P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin. Club: A contrastive log-ratio upper bound of mutual information. In *ICML*, pages 1779–1788. PMLR, 2020a.

P. Cheng, M. R. Min, D. Shen, C. Malon, Y. Zhang, Y. Li, and L. Carin. Improving disentangled text representation learning with information-theoretic guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7530–7541. ACL, 2020b.

Y. Cheng, W. Xu, Z. He, W. He, H. Wu, M. Sun, and Y. Liu. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1965–1974. ACL, 2016.

P. Colombo, P. Piantanida, and C. Clavel. A novel estimator of mutual information for learning to disentangle textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6539–6550. ACL, 2021.

R. Cotterell and J. Kreutzer. Explaining and generalizing back-translation through wake-sleep. In *arXiv preprint arXiv:1806.04402*, 2018.

M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380. ACL, 2014.

O. Domingo, M. R. Costa-jussà, and C. Escolano. A multi-task semi-supervised framework for text2graph & graph2text, 2022.

O. Dušek, D. M. Howcroft, and V. Rieser. Semantic Noise Matters for Neural Natural Language Generation. In *Proceedings of the 12th International Conference on Natural Language Generation (INLG 2019)*, pages 421–426, 2019.

H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. 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 240–250. ACL, 2019.

C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133. ACL, 2017a.

C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini. Creating training corpora for nlg microplanners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188. ACL, 2017b.

S. Gehrmann, T. P. Adewumi, K. Aggarwal, P. S. Ammanamanchi, A. Anuoluwapo, A. Bosselut, K. R. Chandu, M. Clinciu, D. Das, K. D. Dhole, W. Du, E. Durmus, O. Dusek, C. Emezue, V. Gangal, C. Garbacea, T. Hashimoto, Y. Hou, Y. Jernite, H. Jhamtani, Y. Ji, S. Jolly, D. Kumar, F. Ladhak, A. Madaan, M. Maddela, K. Mahajan, S. Mahamood, B. P. Majumder, P. H. Martins, A. McMillan-Major, S. Mille, E. van Miltenburg, M. Nadeem, S. Narayan, V. Nikolaev, R. A. Niyongabo, S. Osei, A. P. Parikh, L. Perez-Beltrachini, N. R. Rao, V. Raunak, J. D. Rodriguez, S. Santhanam, J. Sedoc, T. Sellam, S. Shaikh, A. Shimorina, M. A. S. Cabezudo, H. Strobelt, N. Subramani, W. Xu, D. Yang, A. Yerukola, and J. Zhou. The GEM benchmark: Natural language generation, its evaluation and metrics. *CoRR*, abs/2102.01672, 2021.

Q. Guo, Z. Jin, N. Dai, X. Qiu, X. Xue, D. Wipf, and Z. Zhang. P^2: A plan-and-pretrain approach for knowledge graph-to-text generation. In *WebNLG+*, pages 100–106. ACL, 2020a.

Q. Guo, Z. Jin, X. Qiu, W. Zhang, D. Wipf, and Z. Zhang. CycleGT: Unsupervised graph-to-text and text-to-graph generation via cycle training. In *WebNLG+*, pages 77–88. ACL, 2020b.

Q. Guo, Z. Jin, Z. Wang, X. Qiu, W. Zhang, J. Zhu, Z. Zhang, and D. Wipf. Fork or fail: Cycle-consistent training with many-to-one mappings. In A. Banerjee and K. Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 1828–1836. PMLR, 2021.

D. Gupta and K. Berberich. JIGSAW: structuring text into tables. In Y. Fang, Y. Zhang, J. Allan, K. Balog, B. Carterette, and J. Guo, editors, *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR*, pages 237–244. ACM, 2019.

D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma. Dual learning for machine translation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett,

editors, *NeurIPS*, volume 29. Curran Associates, Inc., 2016.

V. C. D. Hoang, P. Koehn, G. Haffari, and T. Cohn. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24. ACL, 2018.

T. Hu, A. Shrivastava, O. Tuzel, and C. Dhir. Unsupervised style and content separation by minimizing mutual information for speech synthesis. *CoRR*, abs/2003.06227, 2020.

Z. Jin, Q. Guo, X. Qiu, and Z. Zhang. GenWiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2398–2409. International Committee on Computational Linguistics, 2020.

M. Kale and A. Rastogi. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102. ACL, 2020.

P. Ke, H. Ji, Y. Ran, X. Cui, L. Wang, L. Song, X. Zhu, and M. Huang. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538. ACL, 2021.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd ICLR, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

K. Kukich. Design of a knowledge-based report generator. In *ACL*, pages 145–150. ACL, 1983.

A. Laha, P. Jain, A. Mishra, and K. Sankaranarayanan. Scalable Micro-planned Generation of Discourse from Structured Data. *Computational Linguistics*, 45(4):737–763, 2020.

G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *ICLR*, 2018.

R. Lebret, D. Grangier, and M. Auli. Neural text generation from structured data with application to the biography domain. In *EMNLP*, pages 1203–1213. ACL, 2016.

J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. Dbpedia, a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

C. Li, X. Gao, Y. Li, B. Peng, X. Li, Y. Zhang, and J. Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699. ACL, 2020.

J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119. ACL, 2016.

R. Li, X. Peng, and C. Lin. On the latent holes of vaes for text generation, 2021.

C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. ACL, 2004.

T. Liu, K. Wang, L. Sha, B. Chang, and Z. Sui. Table-to-text generation by structure-aware seq2seq learning. *arXiv preprint arXiv:1711.09724*, 2017.

T. Liu, F. Luo, Q. Xia, S. Ma, B. Chang, and Z. Sui. Hierarchical encoder with auxiliary supervision for neural table-to-text generation: Learning better representation for tables. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6786–6793, 2019.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

K. McKeown. *Text Generation*. Studies in Natural Language Processing. Cambridge University Press, 1985.

A. Moryossef, Y. Goldberg, and I. Dagan. 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. ACL, 2019.

D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

L. Nan, D. Radev, R. Zhang, A. Rau, A. Sivaprasad, C. Hsieh, X. Tang, A. Vyas, N. Verma, P. Krishna, Y. Liu, N. Irwanto, J. Pan, F. Rahman, A. Zaidi, M. Mutuma, Y. Tarabar, A. Gupta, T. Yu, Y. C. Tan, X. V. Lin, C. Xiong, R. Socher, and N. F. Rajani. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447. ACL, 2021.

L. Nan, C. Hsieh, Z. Mao, X. V. Lin, N. Verma, R. Zhang, W. Kryściński, H. Schoelkopf, R. Kong, X. Tang, M. Mutuma, B. Rosand, I. Trindade, R. Bandaru, J. Cunningham, C. Xiong, D. Radev, and D. Radev. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49, 2022.

W. K. Ng, G. Yan, and E.-P. Lim. Heterogeneous product description in electronic commerce. *SIGecom Exch.*, 2000.

J. Novikova, O. Dušek, and V. Rieser. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206. ACL, 2017.

G. Paolini, B. Athiwaratkun, J. Krone, J. Ma, A. Achille, R. Anubhai, C. N. dos Santos, B. Xiang, and S. Soatto. Structured prediction as translation between augmented natural languages. In *9th ICLR, ICLR 2021*, 2021.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. ACL, 2002.

A. Parikh, X. Wang, S. Gehrmann, M. Faruqui, B. Dhingra, D. Yang, and D. Das. ToTTo: A controlled table-to-text generation dataset. In *EMNLP*, pages 1173–1186. ACL, 2020.

M. Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. ACL, 2018.

R. Puduppully, L. Dong, and M. Lapata. Data-to-text generation with content selection and planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6908–6915, 2019. doi: 10.1609/aaai.v33i01.33016908.

R. Puduppully, Y. Fu, and M. Lapata. Data-to-text generation with variational sequential planning. *Transactions of the Association for Computational Linguistics*, 10:697–715, 2022.

D. Putthividhya and J. Hu. Bootstrapped named entity recognition for product attribute extraction. In *EMNLP*, pages 1557–1567. ACL, 2011.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (140):1–67, 2020.

C. Rebuffel, L. Soulier, G. Scoutheeten, and P. Gallinari. Capturing Entity Hierarchy in Data-to-Text Generative Models. In *First Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020)*, volume 2621 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR, 2014.

L. F. R. Ribeiro, M. Schmitt, H. Schütze, and I. Gurevych. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227. ACL, 2021.

E. H. Sanchez, M. Serrurier, and M. Ortner. Learning disentangled representations via mutual information estimation, 2019.

M. Schmitt, S. Sharifzadeh, V. Tresp, and H. Schütze. An unsupervised joint system for text generation from knowledge graphs and semantic parsing. In *EMNLP*, pages 7117–7130. ACL, 2020.

D. R. Scott, C. Hallett, and R. Fettiplace. Data-to-text summarisation of patient records: Using computer-generated summaries to access patient histories. *Patient Education and Counseling*, 92:153 – 159, 2013.

C. Shu, Y. Zhang, X. Dong, P. Shi, T. Yu, and R. Zhang. Logic-consistency text generation from semantic parses. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4414–4426. ACL, 2021.

L. Song and D. Gildea. SemBleu: A robust metric for AMR parsing evaluation. In *ACL*, pages 4547–4552. ACL, 2019.

A. Talmor and J. Berant. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921. ACL, 2019.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57 (10):78–85, 2014. ISSN 0001-0782.

J. Wang, Y. Hou, J. Liu, Y. Cao, and C.-Y. Lin. A statistical framework for product description generation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 187–192, Taipei, Taiwan, Nov. 2017. Asian Federation of Natural Language Processing.

Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In J. G. Dy and A. Krause, editors, *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 5167–5176. PMLR, 2018.

S. Wiseman, S. Shieber, and A. Rush. Challenges in data-to-document generation. In *EMNLP*, pages 2253–2263. ACL, 2017.

S. Wiseman, S. Shieber, and A. Rush. Learning neural templates for text generation. In *EMNLP*, pages 3174–3187. ACL, 2018.

T. Xie, C. H. Wu, P. Shi, R. Zhong, T. Scholak, M. Yasunaga, C.-S. Wu, M. Zhong, P. Yin, S. I. Wang, V. Zhong, B. Wang, C. Li, C. Boyle, A. Ni, Z. Yao, D. Radev, C. Xiong, L. Kong, R. Zhang, N. A. Smith, L. Zettlemoyer, and T. Yu. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*, 2022.

P. Xu, J. C. K. Cheung, and Y. Cao. On variational learning of controllable representations for text without supervision. In *ICML*. PMLR, 2020.

H. Ye, N. Zhang, S. Deng, M. Chen, C. Tan, F. Huang, and H. Chen. Contrastive triple extraction with generative transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14257–14265, 2021.

R. Ye, W. Shi, H. Zhou, Z. Wei, and L. Li. Variational template machine for data-to-text generation. In *ICLR*, 2020.

C. Zhao, M. Walker, and S. Chaturvedi. Bridging the structural gap between encoding and decoding for data-to-text generation. In *ACL*, pages 2481–2491. ACL, 2020.

S. Zhao, J. Song, and S. Ermon. Infovae: Information maximizing variational autoencoders. *CoRR*, abs/1706.02262, 2017.

Y. Zhu, S. Lu, L. Zheng, J. Guo, W. Zhang, J. Wang, and Y. Yu. Texygen: A benchmarking platform for text generation models. *SIGIR*, 2018.

A APPENDIX

A.1 Log-likelihood of the Latent Variable Model

Given any arbitrary observed text $x_n^{(t)}$, we can express its log-likelihood using the missing data distribution $p_{\theta^{(d)}}(y_n^{(t)})$ as

$$
\begin{aligned}
\log p_\theta(x_n^{(t)}) &= \underset{p_{\theta^{(d)}}(y_n^{(t)})}{\mathbb{E}}\left[\log p_\theta(x_n^{(t)})\right] \\
&= \underset{p_{\theta^{(d)}}(y_n^{(t)})}{\mathbb{E}}\left[\log \frac{p_{\theta^{(t)}}(x_n^{(t)}|y_n^{(t)})p_{\theta^{(d)}}(y_n^{(t)})}{p_{\theta^{(d)}}(y_n^{(t)}|x_n^{(t)})}\right] \\
&= \underset{p_{\theta^{(d)}}(y_n^{(t)})}{\mathbb{E}}\left[\log p_{\theta^{(t)}}(x_n^{(t)}|y_n^{(t)})\right] + D_{KL}(p_{\theta^{(d)}}(y_n^{(t)})\|p_{\theta^{(d)}}(y_n^{(t)}|x_n^{(t)})).
\end{aligned}
\tag{16}
$$

Likewise, given any observed data $y_m^{(d)}$, for any arbitrary density $p_{\theta^{(t)}}(x_m^{(d)})$ we have

$$
\log p_\theta(y_m^{(d)}) = \underset{p_{\theta^{(t)}}(x_m^{(d)})}{\mathbb{E}}\left[\log p_{\theta^{(d)}}(y_m^{(d)}|x_m^{(d)})\right] + D_{KL}(p_{\theta^{(t)}}(x_m^{(d)})\|p_{\theta^{(t)}}(x_m^{(d)}|y_m^{(d)})).
\tag{17}
$$

A.2 Connection between the Cycle Objective and the Log-likelihood

We use the following notation: $\mathbf{X}^{(t)} = \{x_1^{(t)}, ..., x_N^{(t)}\}$ and $\mathbf{Y}^{(d)} = \{y_1^{(d)}, ..., y_M^{(d)}\}$ are the sets of observed texts and structured data, respectively. $\mathbf{X}^{(d)} = \{x_1^{(d)}, ..., x_N^{(d)}\}$ and $\mathbf{Y}^{(t)} = \{y_1^{(t)}, ..., y_M^{(t)}\}$ are the sets of missing texts and structured data, respectively. We denote by $\theta = (\theta^{(t)}, \theta^{(d)})$ the set of parameters of the generative model. Considering the probablistic graphical model (PGM) in Fig. 2, we have

$$
\begin{aligned}
\log p_\theta(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}) &= \sum_N \log p_{\theta^{(t)}}(x_n^{(t)}) + \sum_M \log p_{\theta^{(d)}}(y_m^{(d)}) \\
&\approx N \underset{x_n^{(t)}\sim p_T(x)}{\mathbb{E}}\left[\log p_{\theta^{(t)}}(x_n^{(t)})\right] + M \underset{y_m^{(d)}\sim p_D(y)}{\mathbb{E}}\left[\log p_{\theta^{(d)}}(y_m^{(d)})\right] \\
&= N \underset{x_n^{(t)}\sim p_T(x)}{\mathbb{E}}\left[\underset{p_{\theta^{(d)}}(y_n^{(t)})}{\mathbb{E}}\left[\log p_{\theta^{(t)}}(x_n^{(t)}|y_n^{(t)})\right] + D_{KL}(p_{\theta^{(d)}}(y_n^{(t)})\|p_{\theta^{(d)}}(y_n^{(t)}|x_n^{(t)}))\right] \\
&\quad + M \underset{y_m^{(d)}\sim p_T(y)}{\mathbb{E}}\left[\underset{p_{\theta^{(t)}}(x_m^{(d)})}{\mathbb{E}}\left[\log p_{\theta^{(t)}}(y_m^{(d)}|x_m^{(d)})\right] + D_{KL}(p_{\theta^{(t)}}(x_m^{(d)})\|p_{\theta^{(t)}}(x_m^{(d)}|y_m^{(d)}))\right]
\end{aligned}
$$

$$\tag{18}$$
$$\tag{19}$$

Iterative back-translation gives us the following:

$$
p_{\theta^{(d)}}(y_n^{(t)}) = p_{\theta^{(d)}}(y_n^{(t)}|x_n^{(t)}),
\tag{20}
$$
$$
p_{\theta^{(t)}}(x_m^{(d)}) = p_{\theta^{(t)}}(x_m^{(d)}|y_m^{(d)}).
\tag{21}
$$

Under these assumptions, the KL terms $D_{KL}(p_{\theta^{(t)}}(x_m^{(d)})\|p_{\theta^{(t)}}(x_m^{(d)}|y_m^{(d)}))$ and $D_{KL}(p_{\theta^{(d)}}(y_n^{(t)})\|p_{\theta^{(d)}}(y_n^{(t)}|x_n^{(t)})))$ vanish. In addition, if $N = M$, then

$$
\begin{aligned}
\underset{\theta}{\operatorname{argmax}} \log p_\theta(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}) &\approx \underset{(\theta^{(t)},\theta^{(d)})}{\operatorname{argmax}} \underset{x_n^{(t)}\sim p_T(x)}{\mathbb{E}}\left[\underset{p_{\theta^{(d)}}(y_n^{(t)}|x_n^{(t)})}{\mathbb{E}}\left[\log p_{\theta^{(t)}}(x_n^{(t)}|y_n^{(t)})\right]\right] \\
&\quad + \underset{y_m^{(d)}\sim p_T(y)}{\mathbb{E}}\left[\underset{p_{\theta^{(t)}}(x_m^{(d)}|y_m^{(d)})}{\mathbb{E}}\left[\log p_{\theta^{(t)}}(y_m^{(d)}|x_m^{(d)})\right]\right] \\
&= \underset{(\theta^{(t)},\theta^{(d)})}{\operatorname{argmax}} \mathcal{L}_{\text{cycle}}(\theta^{(t)}, \theta^{(d)})
\end{aligned}
\tag{22}
$$

Thus, optimizing the cycle objective amounts to optimizing the log-likelihood.

A.3 Derivation of the ELBO

We use the following notation: $\mathbf{X}^{(t)} = \{x_1^{(t)}, ..., x_N^{(t)}\}$ and $\mathbf{Y}^{(d)} = \{y_1^{(d)}, ..., y_M^{(d)}\}$ are the sets of observed texts and structured data, respectively. $\mathbf{X}^{(d)} = \{x_1^{(d)}, ..., x_N^{(d)}\}$ and $\mathbf{Y}^{(t)} = \{y_1^{(t)}, ..., y_M^{(t)}\}$ are the sets of missing texts and structured data, respectively. $\mathbf{C}^{(t)} = \{c_1^{(t)}, ..., c_N^{(t)}\}$ and $\mathbf{C}^{(d)} = \{c_1^{(d)}, ..., c_M^{(d)}\}$ are the set of variables that encode the content from the missing text and structured data dataset, respectively. $\mathbf{S}^{(t)} = \{s_1^{(t)}, ..., s_N^{(t)}\}$ and $\mathbf{S}^{(d)} = \{s_1^{(d)}, ..., s_M^{(d)}\}$ are the set of latent variables that encode the styles of the text in $\mathbf{X}^{(t)}$ and $\mathbf{X}^{(d)}$, respectively. f is the structured data format common to all $\mathbf{Y}^{(d)}$. We denote by $\theta = (\theta^{(t)}, \theta^{(d)})$ the set of parameters of the generative model and by ϕ the parameters of the variational posterior q_ϕ.

The joint likelihood of the model can be expressed as

$$
p_\theta(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}, \mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}; f) = \prod_{n=1}^{N} p(s_n^{(t)}) p_{\theta^{(t)}}(x_n^{(t)}|s_n^{(t)}, c_n^{(t)}) p_{\theta^{(d)}}(y_n^{(t)}; f)
$$
$$
\prod_{m=1}^{M} p(s_m^{(d)}) p_{\theta^{(t)}}(x_m^{(d)}|s_m^{(d)}) p_{\theta^{(d)}}(y_m^{(d)}|c_m^{(d)}; f). \tag{23}
$$

Our goal is to maximize the marginal likelihood (evidence) $p_\theta(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}; f)$ of the observed datasets. We can derive the Evidence Lower Bound (ELBO) as

$$
\log p_\theta(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}; f) = \log \int p_\theta(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}, \mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}; f) d\mathbf{X}^{(d)} d\mathbf{Y}^{(t)} d\mathbf{S}^{(t)} d\mathbf{S}^{(d)}
$$
$$
= \log \mathbb{E}_{q_\phi(\mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}|\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}; f)} \left[\frac{p_\theta(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}, \mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}; f)}{q_\phi(\mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}|\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}; f)} \right]
$$
$$
\geq \mathbb{E}_{q_\phi(\mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}|\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}; f)} \left[\log p_\theta(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}|\mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}; f) \right]
$$
$$
- D_{KL}\left(q_\phi(\mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}|\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}; f) \| p_\theta(\mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}; f) \right)
$$
$$
\text{via Jensen's inequality} \tag{24}
$$
$$
= \mathcal{L}_{(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)})}(\theta, \phi). \tag{25}
$$

Let us now factorize the different terms. Given the PGM in Fig. 3, the likelihood can be factorized as

$$
p_\theta(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}|\mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}; f) = \prod_N p_{\theta^{(t)}}(x_n^{(t)}|s_n^{(t)}, \text{Enc}_\phi(y_n^{(t)})) \prod_M p_{\theta^{(d)}}(y_m^{(d)}|\text{Enc}_\phi(x_m^{(d)}); f). \tag{26}
$$

The prior can be factorized as

$$
p_\theta(\mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}; f) = \prod_N p(s_n^{(t)}) p_{\theta^{(d)}}(y_n^{(t)}; f) \prod_M p(s_m^{(d)}) p_{\theta^{(t)}}(x_m^{(d)}|s_m^{(d)}). \tag{27}
$$

And the variational posterior as

$$
q_\phi(\mathbf{X}^{(d)}, \mathbf{Y}^{(t)}, \mathbf{S}^{(t)}, \mathbf{S}^{(d)}|\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}; f) = \prod_N q_\phi(s_n^{(t)}|x_n^{(t)}) q_\phi(y_n^{(t)}|x_n^{(t)}; f) \prod_M q_\phi(s_m^{(d)}|y_m^{(d)}) q_\phi(x_m^{(d)}|s_m^{(d)}; y_m^{(d)}). \tag{28}
$$

Note that Eq. 24 holds for any arbitrary q_ϕ; different factorizations of the variational posteriors are possible, for instance $q_\phi(\mathbf{X}^{(d)}, \mathbf{S}^{(d)}|\mathbf{X}^{(t)}, \mathbf{Y}^{(d)}) = q_\phi(\mathbf{X}^{(d)}|\mathbf{Y}^{(d)}) q_\phi(\mathbf{S}^{(d)}|\mathbf{X}^{(d)})$, which would result in encoding the style with the synthetic text. We ran experiments with this variant and found that the model performs the best with Eq. 28.

Following Eqs. 26, 27, and 28, The ELBO can be factorized as

$$
\mathcal{L}_{(\mathbf{X}^{(t)}, \mathbf{Y}^{(d)})}(\theta, \phi) = \sum_N \mathcal{L}_{x_n^{(t)}}(\theta^{(t)}, \phi) + \sum_M \mathcal{L}_{y_m^{(d)}}(\theta^{(d)}, \phi), \tag{29}
$$

where each term can be written as

$$\mathcal{L}_{x_n^{(t)}}(\theta^{(t)}, \phi) = \mathbb{E}_{q_\phi(y_n^{(t)}|x_n^{(t)};f)} \left[\mathbb{E}_{q_\phi(s_n^{(t)}|x_n^{(t)})} [\log p_{\theta^{(t)}}(x_n^{(t)}|s_n^{(t)}; \mathrm{Enc}_\phi(y_n^{(t)}))] \right]$$
$$- D_{KL}(q_\phi(y_n^{(t)}|x_n^{(t)};f)\|p_{\theta^{(d)}}(y_n^{(t)};f)) - D_{KL}(q_\phi(s_n^{(t)}|x_n^{(t)})\|p(s_n^{(t)})), \tag{30}$$

$$\mathcal{L}_{y_m^{(d)}}(\theta^{(d)}, \phi) = \mathbb{E}_{q_\phi(s_m^{(d)}|y_m^{(d)})} \left[\mathbb{E}_{q_\phi(x_m^{(d)}|s_m^{(d)};y_m^{(d)})} [\log p_{\theta^{(d)}}(y_m^{(d)}|\mathrm{Enc}_\phi(x_m^{(d)}); f)] - D_{KL}(q_\phi(x_m^{(d)}|y_m^{(d)})\|p_{\theta^{(t)}}(x_m^{(d)}|s_m^{(d)})) \right]$$
$$- D_{KL}(q_\phi(s_m^{(d)}|y_m^{(d)})\|p(s_m^{(d)})). \tag{31}$$

Given our choices of parametrization and prior, we have $q_\phi(s_m^{(d)}|y_m^{(d)}) = p(s_m^{(d)}) = \mathcal{N}(0, I)$. As a consequence, the KL term $D_{KL}(q_\phi(s_m^{(d)}|y_m^{(d)})\|p(s_m^{(d)}))$ vanishes.

We follow the iterative back-translation scheme in Hoang et al. (2018) and consider the prior for the missing text (resp. data) distribution to be the conditional likelihood of the observed text (resp. data), i.e.,

$$p_{\theta^{(d)}}(y_n^{(t)}; f) = p_{\theta^{(d)}}(y_n^{(t)}|\mathrm{Enc}_\phi(x_n^{(t)}); f),$$
$$p_{\theta^{(t)}}(x_m^{(d)}|s_m^{(d)}) = p_{\theta^{(t)}}(x_m^{(d)}|s_m^{(d)}, \mathrm{Enc}_\phi(y_m^{(d)})). \tag{32}$$

However, there is still one major obstacle that prevents us from directly optimizing these lower bounds. Indeed, the operations $y_n^{(t)} \sim q_\phi(y_n^{(t)}|f; x_n^{(t)})$ and $x_m^{(d)} \sim q_\phi(x_m^{(d)}|s_m^{(d)}; y_m^{(d)})$ which correspond to the generation of synthetic training data for back-translation, is a sequence of discrete latent variables, and as a consequence cannot be easily reparametrized. To derive a tractable objective, we will assume that while we optimize for the text likelihood, the variational posterior is a good approximation of the conditional likelihood on the observed data, and vice-versa:

$$q_\phi(y_n^{(t)}|x_n^{(t)}; f) \approx p_{\theta^{(d)}}(y_n^{(t)}|\mathrm{Enc}_\phi(x_n^{(t)}); f), \tag{33}$$
$$q_\phi(x_m^{(d)}|s_m^{(d)}, y_m^{(d)}) \approx p_{\theta^{(t)}}(x_m^{(d)}|s_m^{(d)}, \mathrm{Enc}_\phi(y_m^{(d)})). \tag{34}$$

This resonates with Cotterell and Kreutzer (2018), who interpreted iterative back-translation as iterations of the Wake-Sleep algorithm, where one alternatively learns a forward model $p_\theta(y|x)$ and a variational approximation $q_\phi(y|x)$. The only difference here is that we only consider the wake phase and choose to skip the sleep phase by assuming that Eqs. 33 and 34 hold. This assumption may be rather strong, but can be justified in the iterative back-translation framework, since both models $p_{\theta^{(t)}}(x|y)$ and $p_{\theta^{(d)}}(y|x)$ are optimized to approximate the true conditional distributions $p_T(x|y)$ and $p_D((y|x)$, in alternate steps. Under these assumptions, the KL terms $D_{KL}(q_\phi(y_n^{(t)}|x_n^{(t)}; f)\|p_{\theta^{(d)}}(y_n^{(t)}; f))$ and $D_{KL}(q_\phi(x_m^{(d)}|y_m^{(d)})\|p_{\theta^{(t)}}(x_m^{(d)}))$ vanish. This results in these tractable bounds:

$$\mathcal{L}_{x_n^{(t)}}(\theta^{(t)}, \phi) \approx \mathbb{E}_{p_{\theta^{(d)}}(y_n^{(t)}|\mathrm{Enc}_\phi(x_n^{(t)}); f)} \left[\mathbb{E}_{q_\phi(s_n^{(t)}|x_n^{(t)})} [\log p_{\theta^{(t)}}(x_n^{(t)}|s_n^{(t)}; \mathrm{Enc}_\phi(y_n^{(t)}))] \right] - D_{KL}(q_\phi(s_n^{(t)}|x_n^{(t)})\|p(s_n^{(t)})), \tag{35}$$

$$\mathcal{L}_{y_m^{(d)}}(\theta^{(d)}, \phi) \approx \mathbb{E}_{p(s_m^{(d)})} \left[\mathbb{E}_{p_{\theta^{(t)}}(x_m^{(d)}|s_m^{(d)}, \mathrm{Enc}_\phi(y_m^{(d)}))} [\log p_{\theta^{(d)}}(y_m^{(d)}|\mathrm{Enc}_\phi(x_m^{(d)}); f)] \right]. \tag{36}$$

Note that synthetic samples $y_n^{(t)} \sim p_{\theta^{(d)}}(y_n^{(t)}|\mathrm{Enc}_\phi(x_n^{(t)}); f)$ and $x_m^{(d)} \sim p_{\theta^{(t)}}(x_m^{(d)}|s_m^{(d)}, \mathrm{Enc}_\phi(y_m^{(d)}))$ used for back-translation are computed without gradients. When optimizing for $\theta^{(t)}$, the parameters $\theta^{(d)}$ resulting from the computation of $x_m^{(d)}$ are not updated, and vice-versa.

A.4 Disentanglement through constraining the amount of information in style variables

Some works have defined a measure of disentanglement between two random variables X and Y as the Mutual Information (MI) between the two:

$$I(X; Y) = \mathbb{E}_{p(X,Y)} \left[\log \frac{p(X, Y)}{p(X)p(Y)} \right] = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

From a MI perspective, disentangling X from Y is achieved by minimizing their MI. While some works have explicitly sought to minimize this quantity or an estimate of it (Sanchez et al., 2019; Cheng et al., 2020a,b; Colombo et al., 2021),

other works (Wang et al., 2018; Chang et al., 2022) have simply chosen to introduce a bottleneck on the capacity of the style encoder, constraining the dimensionality of one of the two variables in order to limit the information flow from one variable to the other. For completeness, we show here that by constraining the dimensionality of s, we can upper-bound the mutual information between the content c and the style variables with a function that scales with the dimensionality.

- For the style of the text, from the definition of MI, as the entropy is always positive, we can upper-bound it by:

$$\begin{aligned}
I(s;c) &\leq H(s) \\
&= H(\mathcal{N}(0, \boldsymbol{I})) \\
&= \frac{d}{2}(1 + \log(2\pi)) + \frac{1}{2}\log|\boldsymbol{I}| = \frac{d}{2}(1 + \log(2\pi)),
\end{aligned} \tag{37}$$

where d denotes the dimension of the multivariate Gaussian. Hence, by limiting the dimension of s, we prevent information flowing from the content variable c to s.

- For the format f, as it is a parameter, the entropy is 0:

$$I(f;c) \leq H(f) = 0. \tag{38}$$

Note that although these bounds are not tight, this motivates the use of a lower dimensional space to encode the style.

A.5 Examples from the "Rule" Noise Functions

T2D

```
Roadside Attractions distributed Super Capers which was also distributed by
the public company Lionsgate.

KG: [HEAD] Attractions [TYPE] distributed [TAIL] Super [HEAD] also [TYPE]
distributed [TAIL] public [HEAD] company [TYPE] has_attribute [TAIL] public

TRIPLESET: Attractions : distributed : Super |
also : distributed : public | company : has_attribute : public

MR: name[Attractions], distributed[Super], name[also],
distributed[public], name[company], has_attribute[public]

Totto: <page_title> </page_title> <section_title> </section_title>
<table> <col_header> Entities </col_header> <col_header> Attractions
</col_header>
<col_header> also </col_header> <col_header> company </col_header>
<cell> Super <col_header> distributed </col_header>
<row_header> Attractions </row_header> </cell>
<cell> public <col_header> distributed </col_header>
<row_header> also </row_header> </cell>
<cell> public <col_header> has_attribute </col_header>
<row_header> company </row_header> </cell> </table>
```

D2T

```
[HEAD] Nord (Year of No Light album) [TYPE] artist [TAIL] Year of No Light
[HEAD] Nord (Year of No Light album) [TYPE] genre [TAIL] Post-metal
[HEAD] Nord (Year of No Light album) [TYPE] record label [TAIL] Crucial Blast
--> Nord (Year of No Light album) artist Year of No Light and
Nord (Year of No Light album) record label Crucial Blast and
Nord (Year of No Light album) genre Post-metal
```

```
ARA Veinticinco de Mayo (V-2) : length : 192000.0 (millimetres) |
ARA Veinticinco de Mayo (V-2) : country : Argentina
--> ARA Veinticinco de Mayo (V-2) length 192000.0 (millimetres) and
ARA Veinticinco de Mayo (V-2) country Argentina


name[The Phoenix], eatType[pub], food[French], priceRange[more than £30],
customer rating[5 out of 5], area[riverside], familyFriendly[no],
near[Crowne Plaza Hotel]
--> The Phoenix eatType pub and The Phoenix priceRange more than £30 and
The Phoenix area riverside and The Phoenix near Crowne Plaza Hotel and
The Phoenix name The Phoenix and The Phoenix customer rating 5 out of 5 and
The Phoenix food French and The Phoenix familyFriendly no


<page_title> Gennady Golovkin vs. Daniel Jacobs </page_title>
<section_title> CompuBox stats </section_title>
<table> <cell> Golovkin <col_header> Fighter </col_header> </cell>
<cell> 231/615 <col_header> Total punches </col_header> </cell>
<cell> Jacobs <col_header> Fighter </col_header> </cell>
<cell> 175/541 <col_header> Total punches </col_header> </cell> </table>
--> Gennady Golovkin vs. Daniel Jacobs and CompuBox stats and Golovkin Fighter
and
231/615 Total punches and Jacobs Fighter and 175/541 Total punches
```

A.6 Implementation details

Adapting T5. We build our model around T5 (Raffel et al., 2020), which has an architecture that is close to the original Transformer architecture (Vaswani et al., 2017).

We fine-tune the pre-trained *t5-small* model to generate both text and data. We specify the target format by either adding the prefix "*Generate text:*" (for $\theta^{(t)}$) or "*Generate data:*" (for $\theta^{(d)}$) to the input sequence given to the encoder.

Encoder-Decoder architectures. For the text style encoder $q_\phi(s|x)$, we add a special token "[STYLE]" to the text sequence that is passed to the encoder. We take the latent representation from the encoder output $h(x)$, and add a linear layer to obtain the parameters of our variational distribution:

$$\begin{cases} \mu(x) = W_\mu h(x) + b_\mu \\ \Sigma(x) = W_\Sigma h(x) + b_\Sigma \end{cases}$$

where $q_\phi(s|x) = \mathcal{N}(s; \mu(x), \Sigma(x))$.

For the structured data format f, we simply encode it using a linear layer via $s_f = W_f f + b_f$, before feeding it along with the encoded content $c = \text{Enc}_\phi(x)$ to the decoder.

KL Vanishing Issue. This issue, also known as KL collapse, is a typical problem encountered when training VAE models. Fitting the variational posterior to the prior $\forall x, q_\phi(z|x) = p_\theta(z))$ to get a zero KL divergence term at the beginning of training is a simple way to minimize the ELBO loss. This results in a local minimum where the decoder is capable of performing effectively without the latent variable, which is typically the case with large language models and auto-regressive decoding (Fu et al., 2019). Many works have suggested fixes for this problem; we found the MMD-VAE approach (Zhao et al., 2017) particularly effective.

In practice, the MMD-VAE approach replaces the KL divergence term $D_{KL}(q_\phi(z|x)\|p_\theta(z))$ with a sample estimate of the Maximum-Mean Discrepancy (MMD) measure between the aggregated posterior $q_\phi(z) = \mathbb{E}_{x \sim p_T(x)}[q_\phi(z|x)]$ and the prior $p_\theta(z)$. We also experimented with the β-VAE and the cyclical schedule proposed in Fu et al. (2019) and used in Li et al. (2020), but preliminary experiments were not as successful.

Hyperparameters All models have been trained with the AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 10^{-4} for 10 epochs. Following Lample et al. (2018) and Schmitt et al. (2020), we set $p_{\text{blank}} = p_{\text{repeat}} = 0.2$, $p_{\text{drop}} = 0.1$, and use all 5 noise functions *swap*, *drop*, *blank*, *repeat*, and *rule*. For inference, we use beam search decoding

Table 2: D2T & T2D Evaluation

	D2T			ENTITY			T2D RELATION			SemBLEU	Error (%)
	BLEU	METEOR	ROUGE-L	Precision	Recall	F1	Precision	Recall	F1		
DART											
T5-Large	50.66	**40.00**	-	-	-	-	-	-	-	-	-
BART-large	48.56	39.00	-	-	-	-	-	-	-	-	-
MS-Sup	**52.613**	38.29	**45.48**	**92.87**	**91.16**	**92.01**	**82.33**	**80.41**	**81.36**	**69.1**	0.2747
SS-Sup	45.163	35.91	43.23	85.66	83.91	84.78	72.46	70.54	71.49	54.5	**0.2354**
MS-UnSupVAE	51.074	**37.47**	**44.77**	**85.17**	**84.11**	**84.64**	**73.18**	**72.12**	**72.65**	**67.73**	**0.2354**
SS-UnSupVAE	39.23	33.35	40.72	73.89	71.29	72.56	57.24	54.86	56.03	39.71	1.02
MS-UnSup	48.435	36.2	43.71	82.98	78.63	80.74	70.12	65.43	67.69	56.6	0.4905
SS-Unsup	39.27	33.19	40.51	73.79	68.96	71.29	57.65	52.8	55.12	42.96	0.706
WEBNLG											
MS-Sup	**52.037**	**35.85**	**39.53**	**73.71**	**73.48**	**73.59**	**48.16**	**46.85**	**47.5**	**47.55**	**1.18**
SS-Sup	50.502	35.58	39.31	71.12	70.57	70.84	46.24	44.71	45.46	39.87	0.5
MS-UnSupVAE	**49.575**	**34.3**	**38.46**	**65.94**	**64.84**	**65.38**	**39.9**	**38.57**	**39.22**	**35.62**	4.834
SS-UnSupVAE	43.73	32.5	37.14	60.19	54.53	57.22	27.59	23.48	25.37	16.09	5.06
MS-UnSup	45.172	31.72	36.81	62.35	58.66	60.45	35.52	32.12	33.73	26.97	**1.237**
SS-Unsup	44.72	32.59	36.25	60.5	54.42	57.3	27.94	23.39	25.47	13.62	3.766
E2E											
MS-Sup	40.679	35.3	41.39	**97.7**	**98.04**	**97.87**	**97.47**	**97.81**	**97.64**	**99.21**	0.1083
SS-Sup	**40.95**	**35.43**	**41.48**	96.64	97.29	96.97	96.27	96.95	96.61	98.85	**0.000**
MS-UnSupVAE	39.33	34.94	41	**86.84**	**89.51**	**88.16**	**86.68**	**89.36**	**88**	95.32	**0.000**
SS-UnSupVAE	39.31	35.02	41.05	86.56	84.6	85.57	86.15	84.22	85.18	93.04	**0.000**
MS-UnSup	**39.529**	**34.95**	**41.28**	86.2	86.67	86.44	85.88	86.37	86.13	95.08	**0.000**
SS-Unsup	39.36	34.79	41.01	85.38	73.44	78.96	84.97	73.10	78.59	80.56	**0.000**
TOTTO											
MS-Sup	**44.196**	**37.32**	**36.65**	57.53	**43.94**	**49.82**	**18.03**	**12.9**	**15.04**	11.9	1.312
SS-Sup	43.74	37.08	36.48	**59.46**	42.57	49.62	17.76	11.59	14.03	12.82	**0.2727**
MS-UnSupVAE	**36.793**	**36.11**	**35.92**	54.57	**41.3**	**47.02**	**16.56**	**11.77**	**13.76**	15.14	1.325
SS-UnSupVAE	32.85	33.83	34.37	**55.28**	35.67	43.36	14.02	8.028	10.21	5.72	**0.5065**
MS-UnSup	35.129	34.92	35.09	54.34	38.33	44.95	15.13	9.827	11.92	10.19	0.8961
SS-Unsup	33.40	34.21	34.61	55.41	36.84	44.26	14.10	8.38	10.52	8.06	0.5714

for D2T and T2D with 5 and 8 beams, respectively. We use the coefficient $\lambda = 10$ for the MMD regularization term and use the traditional Gaussian kernel.

A.7 More examples of generated D2T

Input:

```
[TABLECONTEXT] : title : American Playhouse | American Playhouse : role : Jed
Jenkins | [TABLECONTEXT] : [title] : Jeff Daniels | American Playhouse : year :
1982
```

Expected Ouput:

```
In 1982, Jed Jenkins in American Playhouse was played by Jeff Daniels.
```

Generated Texts:

```
Jed Jenkins was a American Playhouse, serving from 1982 to 1982.
The American Playhouse is a play from Jed Jenkins to be cast in 1982.
Jed Jenkins was the American Playhouse, which will be part of American Playhouse
and starring Jeff Daniels.
Jed Jenkins was a member of American Playhouse, served as Jed Jenkins from 1982
to 1982.
Jed Jenkins was the American Playhouse, a place to play again in 1982 and
nominated Jeff Daniels
Jed Jenkins was a play from American Playhouse, serving as Jeff Daniels
Jed Jenkins is a playback that was chosen to be part of American Playhouse in
1982.
```

Jed Jenkins is an American Playhouse, which will be part of the United States House of Representatives.
Jed Jenkins was a playback from American Playhouse, serving another role in 1982.
Jed Jenkins is a Playhouse which was part of American Playhouse, originally cast as Jeff Daniels.

Input:

[HEAD] It's Great to Be Young (1956 film) [TYPE] starring [TAIL] Cecil Parker
[HEAD] Cecil Parker [TYPE] birth year [TAIL] 1897

Expected Outputs:

Cecil Parker, born in 1897, stars in the 1956 film It's Great to be Young.
Cecil Parker, who was born in 1897, starred in the 1956 movie It's Great to Be Young.
It's Great to Be Young starred Cecil Parker, who was born in 1897.

Generated Texts:

It's Great to Be Young starred Cecil Parker in both films,
It's Great to Be Young starred as Cecil Parker, who was born in 1897
It's Great to Be Young starred Cecil Parker, in 1897
It's Great to Be Young are Cecil Parker's films, starting in 1897
It's Great to Be Young starred as Cecil Parker, born in 1897
It's Great to Be Young starred Cecil Parker, both in 1897

Input

<page_title> List of ambassadors of the United States to Trinidad and Tobago </page_title> <section_title> List of ambassadors </section_title> <table> <cell> John L. Estrada <col_header> Name </col_header> </cell> </table>

Expected Output

John Learie Estrada is the United States Ambassador to Trinidad and Tobago.
John L. Estrada is the United States Ambassador to Trinidad and Tobago.
John Learie Estrada is the United States Ambassador to Trinidad and Tobago.

Generated texts:

John L. Estrada was an Ambassador from Trinidad and Tobago, which would be part of the United States to Trinidad and Tobago.
John L. Estrada was an ambassador of the United States to Trinidad and Tobago, serving as Trinidad and Tobago.
John L. Estrada was an Ambassador to Trinidad and Tobago, which was chosen to be part of the United States ambassador to Trinidad and Tobago
John L. Estrada was an American Ambassador to Trinidad and Tobago, coming from another country in Trinidad and Tobago.
John L. Estrada was an Ambassador to Trinidad and Tobago, part of the United States delegation to Trinidad and Tobago.
John L. Estrada was an Ambassador to Trinidad and Tobago, which also served as United States Ambassador to Trinidad and Tobago.
John L. Estrada was the United States Ambassador to Trinidad and Tobago, which was served as part of the United States ambassador to Trinidad and Tobago.
John L. Estrada is an American Ambassador to Trinidad and Tobago, which won its place nationwide.
John L. Estrada was an American Ambassador to Trinidad and Tobago, who served as the United States ambassador to Trinidad and Tobago.
John L. Estrada was an Ambassador to Trinidad and Tobago, who served as the

United States ambassador to Trinidad and Tobago.

Input:

```
name[Blue Spice], eatType[coffee shop], customer rating[5 out of 5], near[Crowne
Plaza Hotel]
```

Expected Outputs:

```
The coffee shop Blue Spice is based near Crowne Plaza Hotel and has a high
customer rating of 5 out of 5.
The Blue Spice coffee shop, near Crowne Plaza Hotel, has a customer rating of 5
out of 5.
If you want a coffee shop rated 5 out of 5 pick Blue Spice. It is located near
Crowne Plaza Hotel.
```

Generated Texts:

```
Blue Spice is a 5 star coffee shop located near Crowne Plaza Hotel that also
serves fast food, not family-friendly.
Blue Spice is a 5-star coffee shop located near Crowne Plaza Hotel that also
offers food in moderate prices.
Blue Spice is a coffee shop located near Crowne Plaza Hotel that offers great
food, at reasonable prices.
Blue Spice is a coffee shop located near Crowne Plaza Hotel that offers cheap
food, but only has a 5 out of 5 rating.
Blue Spice is a 5 star coffee shop located near Crowne Plaza Hotel that also
serves food in cheap prices.
Blue Spice is a coffee shop near Crowne Plaza Hotel that also offers great food,
prices are rated 5 out of 5.
Blue Spice is a 5-star coffee shop located near Crowne Plaza Hotel, that sells
sushi for more than £30.
Blue Spice is a coffee shop located near Crowne Plaza Hotel that sells cheap
food, and rated 5 out of 5.
Blue Spice is a 5-star coffee shop located near Crowne Plaza Hotel that serves
Italian food, and has a price range of more than 30.
Blue Spice is a coffee shop located near Crowne Plaza Hotel that also serves
Japanese food, and has a 5 out of 5 rating.
```