Semantic Matching for Sequence-to-Sequence Learning

Ruiyi Zhang[◇], **Changyou Chen**[†], **Xinyuan Zhang**[‡], **Ke Bai**[◇], **Lawrence Carin**[◇]
[◇]Duke University, [†]State University of New York at Buffalo, [‡]ASAPP Inc.
ryzhang.cs@gmail.com

Abstract

In sequence-to-sequence models, classical optimal transport (OT) can be applied to semantically match generated sentences with target sentences. However, in non-parallel settings, target sentences are usually unavailable. To tackle this issue without losing the benefits of classical OT, we present a semantic matching scheme based on the Optimal Partial Transport (OPT). Specifically, our approach partially matches semantically meaningful words between source and partial target sequences. To overcome the difficulty of detecting active regions in OPT (corresponding to the words needed to be matched), we further exploit prior knowledge to perform partial matching. Extensive experiments are conducted to evaluate the proposed approach, showing consistent improvements over sequence-to-sequence tasks.

1 Introduction

Sequence-to-sequence (Seq2Seq) models are widely used in various natural-language-processing tasks, such as machine translation (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015), text summarization (Rush et al., 2015; Chopra et al., 2016) and image captioning (Vinyals et al., 2015; Xu et al., 2015). Typically, these models are based on an encoder-decoder architecture, with an encoder mapping a source sequence into a latent vector, and a decoder translating the latent vector into a target sequence. The goal of a Seq2Seq model is to optimize this encoder-decoder network to generate sequences close to the target. Therefore, a proper measure of the distance between sequences is crucial for model training.

Wasserstein distance between two text sequences, *i.e.*, word-mover distance (Kusner et al., 2015), can serve as an effective regularizer for semantic matching in Seq2Seq models (Chen et al., 2019). Classical optimal transport models require that each piece of mass in the source distribution is transported to an equal-weight piece of mass in the target distribution. However, this requirement is too restrictive for Seq2Seq models, making direct applications inappropriate due to the following: (*i*) texts often have different lengths, and not every element in the source text corresponds an element in the target text. A good example is style transfer, where some words in the source text do not have corresponding words in the target text. (*ii*) it is reasonable to semantically match *important* words while neglecting some other words, *e.g.*, conjunction. In typical unsupervised models, text data are usually non-parallel in the sense that pairwise data are typically unavailable (Sutskever et al., 2014). Thus, both pairwise information inference and text generation must be performed in the same model with only non-parallel data. Classical OT is not applicable without target text sequences. However, partial target information is available, for example, the detected objects in an image should be described in its caption, and the content words when changing the style should be preserved. OT will fail in these cases but matching can be performed by optimal partial transport (OPT). Specifically, we exploit the partial target information representation via partially matching it with generated texts. The partial matching is implemented based on lexical information extracted from the texts. We call our method SEmantic PArtial Matching (SEPAM).

To demonstrate the effectiveness of SEPAM, we consider applying it on sequence-prediction tasks where semantic partial matching is needed: (*i*) in unsupervised text-style transfer, SEPAM can be employed for content preservation via partially matching the input and generated text; (*ii*) in image captioning, SEPAM can be applied to partially match the objects detected in images with corresponding captions for more informative generation; (*iii*) in table-to-text generation, SEPAM can prevent hallu-

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 212–222

cination (Dhingra et al., 2019) via partially matching tabular key words with generated sentences.

The main contributions of this paper are summarized as follows: (i) A novel semantic matching scheme based on optimal partial transport is proposed. (ii) Our model can be interpreted as incorporating prior knowledge into the optimal transport to exploit the structure of natural language, while making the algorithm tractable for real-world tasks. (iii) In order to demonstrate the versatility of the proposed scheme, we empirically show consistent improvements in style transfer for content preservation, image captioning for informative image descriptions and in table-to-text generation for faithful generation.

2 Background

2.1 Optimal Transport

Optimal transport defines distances between probability measures on a domain $\mathbb{X}$ (the word-embedding space in our setting). The *optimal transport distance* for two probability measures $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ is defined as (Peyré et al., 2017):

$$\mathcal{D}_c(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\gamma \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \gamma} \left[c(\boldsymbol{x}, \boldsymbol{y}) \right], \quad (1)$$

where $\Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$ denotes the set of all joint distributions $\gamma(\boldsymbol{x}, \boldsymbol{y})$ with marginals $\boldsymbol{\mu}(\boldsymbol{x})$ and $\boldsymbol{\nu}(\boldsymbol{y})$; $c(\boldsymbol{x}, \boldsymbol{y}) : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ is the cost function for moving $\boldsymbol{x}$ to $\boldsymbol{y}$, *e.g.*, the Euclidean or cosine distance. Intuitively, the optimal transport distance is the minimum cost that γ induces in order to transport from $\boldsymbol{\mu}$ to $\boldsymbol{\nu}$. When $c(\boldsymbol{x}, \boldsymbol{y})$ is a metric on $\mathbb{X}$, $\mathcal{D}_c(\boldsymbol{\mu}, \boldsymbol{\nu})$ induces a proper metric on the space of probability distributions supported on $\mathbb{X}$, commonly known as the Wasserstein distance (Villani, 2008).

We focus on applying the OT distance on textual data. Therefore, we only consider OT between discrete distributions. Specifically, consider two discrete distributions $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbf{P}(\mathbb{X})$, which can be presented as $\boldsymbol{\mu} = \sum_{i=1}^{n} \mathrm{u}_i \delta_{\boldsymbol{x}_i}$ and $\boldsymbol{\nu} = \sum_{j=1}^{m} \mathrm{v}_j \delta_{\boldsymbol{y}_j}$ with $\delta_{\boldsymbol{x}}$ the Dirac function centered on $\boldsymbol{x}$. The weight vectors $\boldsymbol{u} = \{\mathrm{u}_i\}_{i=1}^{n} \in \Delta_n$ and $\boldsymbol{v} = \{\mathrm{v}_i\}_{i=1}^{m} \in \Delta_m$ belong to the simplex, *i.e.*, $\sum_{i=1}^{n} \mathrm{u}_i = \sum_{j=1}^{m} \mathrm{v}_j = 1$, as both $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are probability distributions. Under such a setting, computing the OT distance defined in (1) can be reformulated as the following minimization problem:

$$\begin{aligned} W_c(\boldsymbol{\mu}, \boldsymbol{\nu}) &= \min_{\mathbf{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{T}_{ij} \cdot c(\boldsymbol{x}_i, \boldsymbol{y}'_j) \\ &= \min_{\mathbf{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \mathbf{T}, \mathbf{C} \rangle , \end{aligned} \quad (2)$$

where $\Pi(\boldsymbol{u}, \boldsymbol{v}) = \{\mathbf{T} \in \mathbb{R}_+^{n \times m} | \mathbf{T}\mathbf{1}_m = \boldsymbol{u}, \mathbf{T}^\top \mathbf{1}_n = \boldsymbol{v}\}$, $\mathbf{1}_n$ denotes an n-dimensional all-one vector, $\mathbf{C}$ is the cost matrix given by $\mathbf{C}_{ij} = c(\boldsymbol{x}_i, \boldsymbol{y}_j)$ and $\langle \mathbf{T}, \mathbf{C} \rangle = \mathrm{Tr}(\mathbf{T}^\top \mathbf{C})$ represents the Frobenius dot-product.

2.2 Optimal Partial Transport

Optimal partial transport (OPT) was studied initially by Caffarelli and McCann (2010). It is a variant of optimal transport, where only a portion of mass is to be transported, in an efficient way. In OPT, the transport problem is defined by generalizing γ as a Borel measure such that:

$$\mathcal{D}_c(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\gamma \in \Pi_{\leq}(f, g),\, \mathcal{M}(\gamma) = m} \int [c(\boldsymbol{x}, \boldsymbol{y})] \mathrm{d}\gamma(\boldsymbol{x}, \boldsymbol{y}) \,. \quad (3)$$

where $\Pi_{\leq}(f, g)$ is defined as the set of nonnegative finite Borel measures on $\mathbb{R}^n \times \mathbb{R}^n$ whose first and second marginals are dominated by f and g respectively, *i.e.*, $\gamma(A \times \mathbb{R}^n) \leq \int_A f(x)dx$ and $\gamma(\mathbb{R}^n \times A) \leq \int_A g(y)dx$ for all $A \in \mathbb{R}^n$; $\mathcal{M}(\gamma) \triangleq \int_{\mathbb{R}^n \times \mathbb{R}^n} d\gamma$ represents the mass of γ in (3), and $m \in [0, \min\{\|f\|_{L_1}, \|g\|_{L_1}\}]$. Here f and g can be considered as the maximum marginal measures for γ. As a result, if m is less than $\min\{\|f\|_{L_1}, \|g\|_{L_1}\}$, this means γ assigns zero measures for some elements of the space. In other words, the zero-measure elements need not be considered when matching $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. The elements with non-zero measure are all active regions. A challenge in OPT is how to detect these active regions. Thus directly optimizing (3) is typically very challenging and computationally expensive. In our setting of text analysis, we propose to leverage prior knowledge to define the active regions, as introduced below.

3 Semantic Matching via OPT

In unsupervised Seq2Seq tasks without pair-wise information, naively matching the generated sequence with the weak-supervision information (*e.g.*, source text in style transfer) will render deficient performance, even though both sentences share similar content. In supervised settings, target and input sequences are of different lengths but have similarity in terms of semantic meaning, such as table-to-text generation. Motivated by this, we propose a novel technique for semantic partial matching and consider two scenarios: (*i*) text-to-text matching and (*ii*) image-to-text matching.

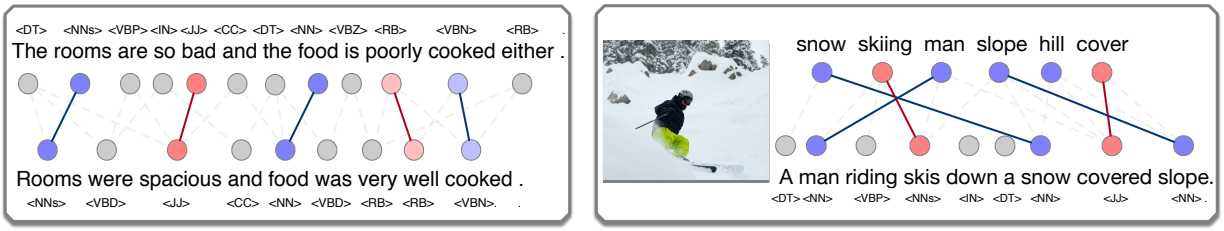

Figure 1: Semantic Matching between the potential target (top) and the generated texts (bottom). The left shows how to partially match two texts with different styles. The right shows how to partially match texts with concepts detected from the images.

Text-to-Text Matching We consider semantic matching between two sequences in Seq2Seq models, where partial matching is important: *i*) in the unsupervised setting, such as non-parallel style transfer, partially matching between the source and target texts is helpful for content preservation. *ii*) in the supervised setting, such as table-to-text generation, partially matching the input and target sequences can effectively avoid hallucination generation, *i.e.* text mentions extra information than what is present in the source. Figure 1 shows an example of partial matching, where part-of-speech (POS) tags for each word are exploited to provide prior knowledge. In these cases, directly applying OT will cause imbalanced transportation issue or poor performance.

Image-to-Text Matching Objects and their properties (*e.g.,* colors) are both included in the pair-wise images and captions. Consider the image-to-text matching in Figure 1. It is clear that each object in the image has corresponding words/phases in the captions. We can consider matching the labels of detected objects in the image to some words in its caption. Please note labels are not in one-to-one correspondence with the text, thus directly applying OT is inappropriate, similar to the case of text-to-text matching.

Different Matching Schemes *Hard matching* seeks to exactly match from the source and target. Typically, hard matching is too simple to be effective without considering semantic similarity, and if we apply classical optimal transport in unsupervised settings, it causes an imbalance matching, since some unnecessary words are included in the source and the exact target is unavailable. To tackle this issue, one can directly apply the optimal partial transport (OPT) here to detect which word has its correspondence and match the word with its target. However, the detection process is computationally expensive, which is not scalable as a constrained optimization in (3) for real-world tasks. Fortunately, we can exploit the syntax information from text, and incorporate this information as prior knowledge into OPT to avoid the detection procedure.

3.1 Partial Matching via OPT

We formulate the proposed semantic matching as a partial optimal-transport problem, where only parts of the source and target are matched. Specifically, we incorporate prior knowledge into the optimal partial transport (OPT), and this prior knowledge helps determine the set of words to match, *i.e.*, $M(\boldsymbol{X})$, where $M(\cdot)$ is a function giving a set including the words/phases to match. The strategy of how to determine $M(\cdot)$ depends on tasks.

OPT distance To apply the OT distance to text, we first represent a sentence $\boldsymbol{Y}$ with a discrete distribution $p_{\boldsymbol{Y}} = \frac{1}{T}\sum_t \delta_{e(y_t)}$ in the semantic space, where the length-normalized point mass is placed at the semantic embedding, $\boldsymbol{e}_t^y = e(y_t)$, of each token y_t of the sequence $\boldsymbol{Y}$. Here $e(\cdot)$ denotes a word-embedding function mapping a token to its d-dimensional feature representation. For two sentences $\boldsymbol{X}$ and $\boldsymbol{Y}$, we define their OPT distance as:

$$W_c(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{T} \in \Pi_c(\boldsymbol{\mu}, \boldsymbol{\nu})} \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{T}_{ij} c(\boldsymbol{e}_i^x, \boldsymbol{e}_j^y) \,, \quad (4)$$

where $\Pi_c(\boldsymbol{\mu}, \boldsymbol{\nu})$ is the solution space, and every solution $\mathbf{T} \in \Pi_c(\boldsymbol{\mu}, \boldsymbol{\nu})$ satisfies $\mathbf{T}_{ij} = 0$ if $x_i \notin M(\boldsymbol{X})$ or $y_j \notin M(\boldsymbol{Y})$. Different from classical OPT, the elements in $\boldsymbol{\mu}$ or $\boldsymbol{\nu}$ to match have been explicitly defined by $M(\cdot)$, which represents the *prior knowledge*. In more detail, the constraint of OPT is more specific, and does not need any optimization procedure. We use *cosine distance* as the cost function and $c(\boldsymbol{e}^x, \boldsymbol{e}^y) \triangleq 1 - \frac{\boldsymbol{e}^{x\,T}\boldsymbol{e}^y}{\|\boldsymbol{e}^x\|_2 \|\boldsymbol{e}^y\|_2}$.

Approximation of OPT Computing the exact OPT distance is computationally challenging (Figalli, 2010). We bypass the difficulty of active region detection using lexical information and reformulate it as an OT problem. We then em-

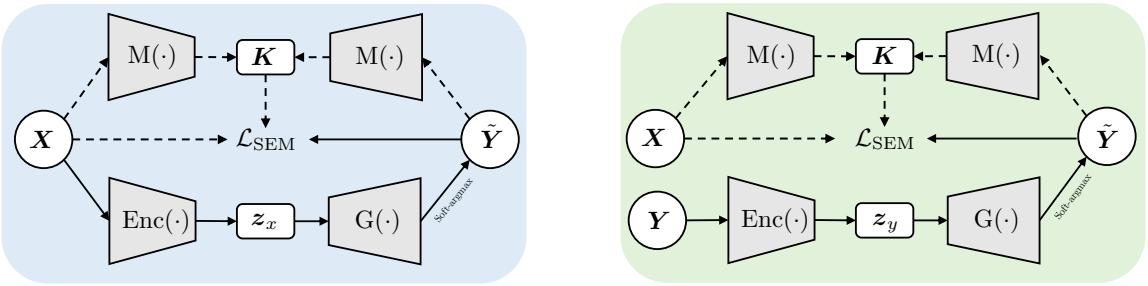

Figure 2: Overview of the SEPAM architecture. Left: classical Seq2Seq, *i.e.*, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are pair-wise; $\mathcal{L}_{\text{SEM}}$ implements a soft-copying mechanism via semantic partial matching. Right: unsupervised Seq2Seq, *i.e.*, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are non-parallel; $\mathcal{L}_{\text{SEM}}$ provides the guidance for $G(\cdot)$, to generate $\tilde{\boldsymbol{Y}}$ relevant to $\boldsymbol{X}$ via semantic partial matching. Solid lines mean gradients are backpropagated in training; dash lines mean gradients are not backpropagated.

ploy the IPOT algorithm (Xie et al., 2018) to obtain an efficient approximation. In practice, we use a keyword mask $\mathbf{K}$, defined as $K_{ij} = 0$ if $x_i \notin M(\boldsymbol{X})$ and $y_j \notin M(\boldsymbol{Y})$; $K_{ij} = +\infty$ if $x_i \notin M(\boldsymbol{X})$ xor $y_j \notin M(\boldsymbol{Y})$; and $K_{ij} = 1$, otherwise. Hence we define the OPT distance as $W_c^p\left(p_{\boldsymbol{X}}, p_{\boldsymbol{Y}}, \mathbf{K}\right) \triangleq W_c\left(p_{M(\boldsymbol{X})}, p_{M(\boldsymbol{Y})}\right)$. Specifically, IPOT considers proximal gradient descent to solve the optimal transport matrix:

$$\mathbf{T}^{(t+1)} = \underset{\mathbf{T}\in\Pi_c(\boldsymbol{\mu},\boldsymbol{\nu})}{\arg\min} \left\{\langle \mathbf{T}, \mathbf{C}'\rangle + \gamma \cdot \mathbb{D}_{\text{KL}}(\mathbf{T}, \mathbf{T}^{(t)})\right\}, \quad (5)$$

where $\mathbf{C}' = \mathbf{K} \circ \mathbf{C}$, $1/\gamma > 0$ is the generalized step-size, and the generalized KL-divergence $\mathbb{D}_{\text{KL}}(\mathbf{T}, \mathbf{T}^{(t)})$ is used as the proximity metric. The full approach is summarized as Algorithm 1 in Appendix A.

4 Semantic Partial Matching for Text Generation

Assume there are two sets of objects $\mathcal{X} = \{\boldsymbol{X}^{(i)}\}_{i=1}^{M}$ and $\mathcal{Y} = \{\boldsymbol{Y}^{(j)}\}_{j=1}^{N}$, we consider a Seq2Seq model, where the input is[1] $\boldsymbol{X}$, and the output is a sequence of length T with tokens y_t, *i.e.*, $\boldsymbol{Y} = (y_1, y_2 \ldots, y_T)$. One typically assigns the following probability to an observation y at location t: $p(y|\boldsymbol{Y}_{<t}) = [\text{softmax}(g(\boldsymbol{s}_t))]_y$, where $\boldsymbol{Y}_{<t} = (y_1, y_2 \ldots, y_t)$. This specifies a probabilistic model, *i.e.*,

$$\log p(\boldsymbol{Y}|\boldsymbol{X}) = \sum_t \log p(y_t|\boldsymbol{Y}_{<t}, \boldsymbol{X}). \quad (6)$$

To train the model, one typically uses maximum likelihood estimation (MLE):

$$\mathcal{L}_{\text{MLE}} = -\mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim(\mathcal{X},\mathcal{Y})}[\log p(\boldsymbol{Y}|\boldsymbol{X})]. \quad (7)$$

[1]For simplicity, we omit the superscript "i" when the context is independent of i. This applies to $\boldsymbol{Y}$.

We consider an encoder-decoder framework in this paper, where a latent vector $\boldsymbol{z}$ is given by an encoder $\text{Enc}(\cdot)$, with input $\boldsymbol{X}$, *i.e.*, $\boldsymbol{z}_x = \text{Enc}(\boldsymbol{X})$. Based on $\boldsymbol{z}_x$, a decoder $G(\cdot)$ generates a new sentence $\hat{\boldsymbol{Y}}$ that is expected to be the same as $\boldsymbol{Y}$. The decoder can be implemented by an LSTM (Hochreiter and Schmidhuber, 1997), GRU (Cho et al., 2014), or Transformer (Vaswani et al., 2017). A unsupervised Seq2Seq model considers $\mathcal{X}$ and $\mathcal{Y}$ as non-parallel, *i.e.*, the pair-wise information is unknown. One typically pretrains the generator with the reconstruction loss:

$$\mathcal{L}_{\text{AE}} = \mathbb{E}_{\boldsymbol{Y}\sim\mathcal{Y}}[-\log p(\boldsymbol{Y}|\boldsymbol{z}_y)], \quad (8)$$

where $\boldsymbol{z}_y = \text{Enc}(\boldsymbol{Y})$. Note the goal of unsupervised Seq2Seq is to generate a sequence $\boldsymbol{Y}$ given some object $\boldsymbol{X}$. Hence, we seek to learn the conditional generation distributions $p(\boldsymbol{Y}|\boldsymbol{X})$, the same as the classical Seq2Seq model. In practice, the generator can be trained combining the reconstruction loss with some guidance loss containing the information from $\boldsymbol{X}$. The guidance loss function can be defined following SEPAM, and the others usually depend on tasks and we omit their details for clarity.

Differentiable SEPAM Note that the SEPAM loss is not differentiable due to the multinomial distribution sampling process $\hat{y}_t \sim \text{Softmax}(\boldsymbol{g}_t)$, where $\boldsymbol{g}_t$ is a logit vector given by the final layer of the generator $G(\cdot)$. To enable direct backpropagation from the SEPAM loss for generator training, we consider the soft-argmax approximation (Zhang et al., 2017) to avoid the use of REINFORCE (Sutton et al., 2000):

$$\tilde{y}_t = \text{Softmax}(\boldsymbol{g}_t/\tau),$$

where $0 < \tau < 1$ is the annealing factor. Given two sentences, we denote the generated sequence

embeddings as $\mathbf{S}_g = \{\tilde{\boldsymbol{e}}_i^y\}_{i=1}^T$ and partial reference embedding as $\mathbf{S}_r = \{\boldsymbol{e}_j^x\}_{j=1}^{T'}$ in word or phrase level. The cost matrix $\mathbf{C}$ is then computed as $C_{ij} = c(\tilde{\boldsymbol{e}}_i^y, \boldsymbol{e}_j^x)$. The semantic partial matching loss between the reference and model generation can be computed via the IPOT algorithm:

$$\mathcal{L}_{\text{SEM}} = W_c^p(\mathbf{S}_g, \mathbf{S}_r, \mathbf{K}) \ . \tag{9}$$

SEPAM Regularization SEPAM training objectives discussed above only focus on generating words with specific meanings and do not consider the word-ordering. To train a proper text-generation model, we propose to combine the SEPAM loss with the likelihood loss $\mathcal{L}_{\text{MLE}}$ in supervised settings or $\mathcal{L}_{\text{AE}}$ in unsupervised settings. Hence, we have the training objective in unsupervised settings: $\mathcal{L} = \mathcal{L}_{\text{AE}} + \lambda\mathcal{L}_{\text{SEM}}$, where λ is the weight of SEPAM to be tuned. A similar objective applies for supervised settings: $\mathcal{L} = \mathcal{L}_{\text{MLE}} + \lambda\mathcal{L}_{\text{SEM}}$. In the following, we discuss how to extract and use prior knowledge for partial matching in three downstream tasks:

(*i*) **Non-parallel Style Transfer** Semantic partial matching between the source sentence and the transferred one is helpful for content preservation, as shown in Figure 1. It is usually the case that the content words are nouns or verbs, and style words are adjectives or adverbs. Hence, $M(\boldsymbol{X})$ and $M(\hat{\boldsymbol{Y}})$ are content word sets, extracted based on the POS tags using NLTK. One can employ this prior knowledge to perform different operations for words: (*i*) for content words, we should encourage partial matching between the sentences by $\mathcal{L}_{\text{SEM}}$; and (*ii*) for style words, we should discourage matching (Hu et al., 2017). More details are provided in Appendix A.1.

(*ii*) **Unsupervised Image Captioning** Visual concepts extracted from an image can be employed for generating relevant captions in the unsupervised setting. Feng et al. (2019) uses exactly hard-matching and REINFORCE to update the captioning model. Here, we apply the semantic partial matching to encourage the generatation of visual-concept words. Specifically, $M(\boldsymbol{X})$ releases the visual concepts and $M(\hat{\boldsymbol{Y}})$ corresponds to the generated words realted to the object (*i.e.*, nouns). This visual concept regularization can also be applied in the supervised setting complementing with MLE loss.

(*iii*) **Table-to-Text Generation** Semantic partial matching can prevent hallucination generation, *i.e.*, text mentions extra information than what is present in the table (Dhingra et al., 2019). $M(\hat{\boldsymbol{Y}})$ extracts nouns from the generated sequence, and $M(\boldsymbol{X})$ are keys in the table, then we used them to compute $\mathcal{L}_{\text{SEM}}$. Semantic partial matching will penalize the generator if extra information exists in the generated text $\hat{\boldsymbol{Y}}$.

5 Related Work

Optimal transport in NLP Kusner et al. (2015) first applied optimal transport in NLP, and proposed the *word mover's distance* (WMD). The transportation cost is usually defined as Euclidean distance, making the OT distance approximated by solving a less-accurate lower bound (Kusner et al., 2015). Based on this, Chen et al. (2018) proposed a feature-mover distance for style transfer. Chen et al. (2019) applied OT for classical seq-to-seq, and formulated it as Wasserstein gradient flows (Zhang et al., 2018). SEPAM moves forward and applies OT in both supervised and unsupervised settings (Artetxe et al., 2018; Lample et al., 2017).

Unsupervised Seq2Seq Learning Different from the standard Seq2Seq model (Sutskever et al., 2014), parallel sentences for different styles are not provided, and must be inferred from the data. Unsupervised machine translation (Artetxe et al., 2018; Lample et al., 2017) learns to translate text from one language to another with two sets of texts of these languages provided. Dai et al. (2019) explores the transformer model as the generator, instead of classical auto-regressive models. Style transfer (Shen et al., 2017) aims at transferring the styles of the texts with non-parallel data. Compared with these tasks, unsupervised image captioning (Feng et al., 2019) is more challenging since images and sentences are in distinct modalities.

Copying Mechanism This is related to the copy network (Gu et al., 2016), which achieves retrieved-based copying. Li et al. (2018) further proposed a delete-retrieve-generate framework for the style transfer. Chen and Bansal (2018) combine the abstraction with extraction in text summarization, and achieves state-of-the-art results via reinforced word selection. In this work, we proposed the semantic partial matching, which can be regarded as a kind

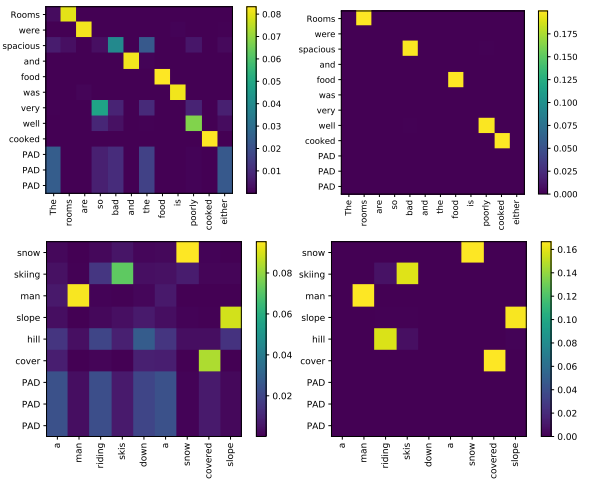

Figure 3: Optimal matching matrix visualization. A comparison between OT (left column) and SEPAM (right column). The Optimal matching matrix of SEPAM is sparse. The horizontal axis are the generated texts, and the vertical axis are the partial targets.

of soft-copying mechanism. Instead of the retrieval-based exact copying used by (Gu et al., 2016; Li et al., 2018), SEPAM considers semantic similarity, and thus ideally delivers smoother transformation in generation.

6 Experiments

6.1 Demonstration

Comparison between OT and SEPAM We show two examples of classical OT and SEPAM under two sequence-prediction tasks in Figure 3. The first row shows the heat map of OT and SEPAM on matching two sentences with different styles. SEPAM employs the syntax information to match selected words and all the content words are exactly matched. However, some sentiment words in classical OT are still matched, preventing successful style transfer. The second row in Figure 3 shows the comparison on matching a generated sentence with the detected concept set in image captioning. The concepts are perfectly matched with their corresponding words in the caption using SEPAM, while OT includes some noisy matching (light blue). In summary, SEPAM achieves better matching than classical OT, and the matching weights ($\mathbf{T}$) of SEPAM is more sparse.

Implicit Use of Prior Knowledge We consider using the weights w_t of attentions from a LSTM-based text classifier to determine which words to match. As discussed in Wiegreffe and Pinter (2019), a word with higher attention weight means it is more important for classification, *i.e.*, more relevant to the style. As shown in Figure 4, shows the attention maps of three instances. Hence, we can partially match words with lower attention weights as they are mostly non-style words. However, empirical results show implicit ways have much worse results than the simple rule-based strategy with POS tags.

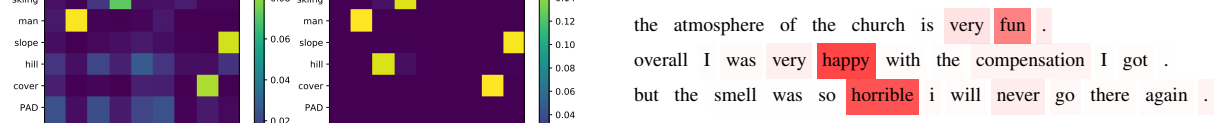

Figure 4: Attention maps for three Yelp instances. Larger attention weight corresponds to darker color.

6.2 Unsupervised Text-style Transfer

Setup We use the same data and split method described in (Shen et al., 2017). Experiments are implemented with Tensorflow based on texar (Hu et al., 2018). For a fair comparison, we use a similar model configuration to that of (Hu et al., 2017; Yang et al., 2018). One-layer GRU (Cho et al., 2014) encoder and LSTM attention decoder (generator) are used. We set the weight of semantic matching loss as $\lambda = 0.2$. Models are trained for a total of 15 epochs, with 10 epochs for pretraining and 5 epochs for fine-tuning.

Metrics We pretrain a CNN classifier, which achieves an accuracy of 97.4% on the validation set. Based on it, we report the accuracy (**ACC**) to measure the quality of style control. We further measure the content preservation using *i*) **BLEU**, which measures the similarity between the original and transferred sentences *ii*) **ref-BLEU**, which measures content preservation comparing the transferred sentences with human annotations. Fluency is evaluated with perplexity (**PPL**) of generated sentences based on a pretrained language model.

Baselines We implemented CtrlGen (Hu et al., 2017) and LM (Yang et al., 2018) as our baselines and further added SEPAM on these two models for validation. Further, conditional variational encoder (CVAE) (Shen et al., 2017) and retrieval-based methods (Li et al., 2018) are added as two baselines.

Analysis Results are shown in Tables 1. It may be observed that combining our proposed method with corresponding baselines exhibits similar transfer accuracy and fluency, while maintaining the content better. Specifically, SEPAM shows higher BLEU scores on human annotated sentences, fur-

Model	ACC (%)	BLEU	ref-BLEU	PPL
CAE (Shen et al., 2017)	73.9	20.7	7.8	51.6
(Fu et al., 2018):				
StyleEmbedding	8.1	**67.4**	19.2	120.1
MultiDecoder	46.9	40.1	12.9	113.1
(Li et al., 2018):				
Template	80.1	57.4	20.5	170.5
DeleteAndRetrieval	88.9	36.8	14.7	74.2
CtrlGen (Hu et al., 2017)	89.0	61.4	22.3	176.8
OT + CtrlGen	85.4	62.9	21.7	183.7
SEPAM+ CtrlGen	89.1	63.7	25.9	176.4
LM (Yang et al., 2018)	88.3	60.5	25.7	79.9
OT + LM	84.8	61.8	22.8	85.1
SEPAM+ LM	88.7	62.0	**28.2**	76.6

Table 1: Our model and baselines performance on test dataset with human annotations.

Input:	tasted really old , i could n't believe it .
CtrlGen:	adds really top , i could gorgeous believe it .
SEPAM+ CtrlGen:	tasted really *surprisingly* , i could fantastic believe it .
LM:	tasted really great , i could *always* believe it .
SEPAM+ LM:	tasted really *excellent* , i could *always* believe it .
Input:	they do not stock some of the most common parts .
CtrlGen:	they do fantastic laughed some of the most common parts .
SEPAM+ CtrlGen:	they do authentic expertly some of the most *fascinating* parts
LM:	they do definitely right some of the most cool parts .
SEPAM+ LM:	they do *always stock* some of the most amazing parts .
Input:	the woman who helped me today was very friendly and knowledgeable .
CtrlGen:	the woman who so-so me today was very rude and knowledgeable .
SEPAM+ CtrlGen:	the woman who *helped* me today was very *rude* and knowledgeable .
LM:	the woman who ridiculous me today was very rude and knowledgeable .
SEPAM+ LM:	the woman who helped me today was very rude and *stupid* .

Table 2: Examples for comparison of different methods on Yelp dataset.

ther validating its effectiveness on content preservation. It is interesting to see that lower BLEU scores with original sentences does not imply higher BLEU scores with the human annotations in Table 1. Compared with other models, the proposed model shows a better balance among accuracy, fluency and content preservation, achieving the highest ref-BLEU.

Model	Style	Content	Fluency
CAE (Shen et al., 2017)	3.21	2.91	2.83
CtrlGen (Hu et al., 2017)	3.42	3.22	2.79
LM (Yang et al., 2018)	3.38	3.32	3.20
SEPAM+ CtrlGen	**3.51**	3.56	2.88
SEPAM+ LM	3.47	**3.72**	**3.25**

Table 3: Human evaluation results on Yelp dataset.

Human Evaluation We further conduct human evaluations for the proposed SEPAM using Amazon Mechnical Turk. We randomly sample 100 sentences from the test set and ask 5 different reviewers to provide their rating scores of the models in terms of fluency, style, and content preservation. We require all the workers to be native English speakers, with approval rate higher than 95% and at least 100 assignments completed. For each sentence, five shuffled samples generated by different models are sequentially shown to a reviewer. Results in Table 3 demonstrate that the better performance achieved by SEPAM, especially in terms of content preservation.

6.3 Image Captioning

Setup We consider image captioning using the COCO dataset (Lin et al., 2014), which contains 123,287 images in total and each image is annotated with at least 5 captions. Following Karpathy's split (Karpathy and Fei-Fei, 2015), 113,287 images are used for training and 5,000 are used for validation and testing. We note that the training images are used to build the image set, *with all the captions left unused for any training*. All the descriptions in the Shutterstock image description corpus are tokenized with a vocabulary size of $18,667$ (Feng et al., 2019). The LSTM hidden dimension and the shared latent space dimension are fixed to 512. The weighting hyper-parameters are chosen to make different rewards roughly the same scale. Specifically, λ is set to 10. We train our model using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001. During the initialization process, we minimize the cross-entropy loss using Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.001. When generating captions in the test phase, we use beam search with a beam size of 3.

Metrics We report BLEU (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), and METEOR (Banerjee and Lavie, 2005) scores. The results of different methods are shown in Table 5.

Method	BLEU	METEOR	CIDEr	SPICE
Feng et al. (2019)	38.2	27.5	22.9	6.6
OUR IMPLEMENTATIONS				
Hard Matching	38.5	27.2	28.2	7.8
OT	39.9	28.1	28.3	7.9
SEPAM	**42.1**	**28.9**	**30.2**	**8.4**

Table 5: Performance comparisons of Unsupervised captioning on the MSCOCO dataset.

Analysis Results in Table 5 show consistent improvement of SEPAM over classical OT. Classical

Method	BLEU	METEOR	ROUGE	PARENT / Precision / Recall
Seq2Seq (Wiseman and Rush, 2016)	22.24	19.50	39.49	43.41 / 49.09 / 41.80
Pointer (See et al., 2017)	19.32	19.88	40.68	49.52 / 61.73 / 44.09
Structre Aware (Liu et al., 2018)	22.76	20.27	39.32	46.47 / 51.18 / 46.34
Transformer	23.48	21.89	42.50	52.60 / 63.20 / 47.90
Transformer + OT	23.87	**22.35**	42.03	51.81 / 60.65 / 48.87
Transformer + SEPAM	**24.06**	22.29	**42.83**	**53.16 / 62.99 / 48.81**

Table 4: Performance comparisons of Table-to-Text Generation on the WikiPerson.

OT can improve upon the baseline via generating specific words aligned with the detected visual concepts. However, directly applying it in unsupervised settings will suffer from the imbalance issue (Craig, 2014), *i.e.*, the generated texts contains some useless elements without correspondence in the targets. Our proposed SEPAM can avoid this problem via partial matching, leading to better performance.

Extension to Supervised Settings Our proposed SEPAM $\mathcal{L}_{\text{SEM}}$ can also be applied in a supervised setting as a regularizer with the MLE loss. We apply SEPAM in the captioning model, where image features are fed into an LSTM sequence generator with an Att2in attention mechanism (Anderson et al., 2018). We pretrain the captioning model for a maximum of 20 epochs, then use reinforcement learning to train it for another 20 epochs. Testing is done on the best model with the validation set. We partially match the tags or visual features of detected objects. Similarly, we see consistent improvement of SEPAM over its baselines.

Method	BLEU	METEOR	ROUGE	CIDEr
Vinyals et al. (2015)	27.7	23.7	-	85.5
Gan et al. (2017)	56.6	25.7	-	101.2
Lu et al. (2017)	33.2	26.6	-	108.5
Chen et al. (2019)	33.8	25.6	-	102.9
OUR IMPLEMENTATIONS				
MLE	34.3	26.2	55.2	106.3
Visual + OT	34.6	26.4	55.6	107.5
Visual + SEPAM	**34.9**	26.9	56.0	109.2
Tag + OT	34.8	26.5	55.6	107.9
Tag + SEPAM	34.4	**27.0**	**56.1**	**111.6**

Table 6: Performance comparisons of supervised image captioning results on the MSCOCO dataset.

6.4 Table-to-Text Generation

Setup We evaluate SEPAM on table-to-text generation (Lebret et al., 2016; Liu et al., 2018; Wiseman et al., 2018) with the WikiPerson dataset (Wang et al., 2018) and preprocess the training set, with a vocabulary size of 50, 000. We use the transformer encoder and decoder. We set the number of heads as 8, the number of Transformer blocks as 3, the hidden units of the feed-forward layer as 2048 and $\lambda = 0.1$. Similarly, the model is first trained with $\mathcal{L}_{\text{MLE}}$ for 20, 000 steps and then fine-tuning with $\mathcal{L}_{\text{SEM}}$.

Slot Name	Slot Value
<Name_ID>	Xia Jin
<occupation>	Football player
<date of birth>	14 February 1985
<place of birth>	Chongqing
<member of sports team>	Guizhou Hengfeng F.C.
<member of sports team>	Chongqing Dangdai Lifan F.C.
<member of sports team>	Chengdu Better City F.C.

MLE: Xia Jin (born 14 February 1985 in Guizhou) is a Chinese Football player who currently plays for Guizhou Dangdai Lifan F.C. in the China League One . he joined Guizhou Dangdai Lifan F.C. in the summer of 2010 . he joined Guizhou Dangdai Lifan F.C. in the summer of 2013 . he joined Guizhou Dangdai Lifan F.C. in the summer of 2013 . he joined Guizhou Dangdai Lifan F.C. in the summer of 2013 . he started his professional career with Chongqing Dangdai Lifan F.C. .

OT: Xia Jin (born 14 February 1985 in *Chongqing*) is a Chinese Football player who currently plays for Guizhou Hengfeng F.C. in the China League One . jin started his professional footballer career with *Guizhou Hengfeng F.C.* in the Chinese Super League . jin would move to China League One side Chongqing Dangdai Lifan F.C. in February 2011 . he would move to China League Two side *Chengdu Better City F.C.* in January 2012 . he would move to China League Two side Chongqing Dangdai Lifan F.C. in January 2013.

SEPAM: Xia Jin (born 14 February 1985 in *Chongqing*) is a Chinese Football player who currently plays for *Guizhou Hengfeng F.C.* in the China League One . Xia Jin started his professional footballer career with *Chongqing Dangdai Lifan F.C.* in the Chinese Super League . Xia transferred to China League One side *Chengdu Better City F.C.* .

Table 7: An example of Table-to-Text Generation.

Metrics For automatic evaluation, we apply the widely used evaluation metrics including the standard BLEU(-4) (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE (Lin, 2015) scores to evaluate the generation quality. Following Dhingra et al. (2019), we evaluate with PARENT score on hallucination generation, which considers both the reference texts and table content in evaluations.

Analysis Results in Table 4 show consistent improvement of SEPAM over baselines in terms of different evaluation metrics. Table 7 shows an example of table-to-text generation. MLE hallucinates some information that does not appear in the

table. OT alleviates this issue, but still shows hallucination since the imbalance transportation issue. SEPAM generates almost no extra information, and covers all the entries in the table.

7 Conclusions

We incorporate prior knowledge into optimal transport, to encourage partial-sentence matching via formulating it as an optimal partial transport problem. The proposed SEPAM shows broad applicability and consistent improvements against popular baselines in three downstream tasks: unsupervised style transfer for content preservation, image captioning for informative descriptions and table-to-text generation for faithful generation. Further, the proposed technique can be regarded as a soft-copying mechanism for Seq2Seq Models.

Acknowledgment The authors would like to thank Zhenyi Wang, Liqun Chen and Siyang Yuan for helpful thoughts and discussions. The Duke University research was funded in part by DARPA, DOE, NSF and ONR.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *ICLR*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*.

Luis Caffarelli and Robert J McCann. 2010. Free boundaries in optimal transport and monge-ampere obstacle problems. *Annals of mathematics*.

Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. 2018. Adversarial text generation via feature-mover's distance. In *NeurIPS*.

Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. 2020. Graph optimal transport for cross-domain alignment. In *ICML*.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Improving sequence-to-sequence learning via optimal transport. In *ICLR*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *ACL*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL*.

Katy Craig, editor. 2014. *The exponential formula for the Wasserstein metric*. PhD thesis, The State University of New Jersey.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. *arXiv preprint arXiv:1905.05621*.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of 57th Annual Meeting of the Association for Computational Linguistics*.

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: Better text generation via filling in the _. In *ICLR*.

Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. 2019. Unsupervised image captioning. In *CVPR*.

Alessio Figalli. 2010. The optimal partial transport problem. *Archive for rational mechanics and analysis*.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *AAAI*.

Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. 2017. Semantic compositional networks for visual captioning. In *CVPR*.

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. In *ACL*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

Zhiting Hu, Haoran Shi, Zichao Yang, et al. 2018. Texar: A modularized, versatile, and extensible toolkit for text generation. *arXiv preprint arXiv:1809.00794*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Controllable text generation. In *ICML*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *ICML*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. In *ICLR*.

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

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *NAACL*.

Chin-Yew Lin. 2015. Rouge: A package for automatic evaluation of summaries.

Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *NIPS*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

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

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *IJCAI*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Gabriel Peyré, Marco Cuturi, et al. 2017. Computational optimal transport. Technical report.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *ACL*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv:1509.00685*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: summarization with pointer-generator networks. In *ACL*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *NIPS*.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. Transforming delete, retrieve, generate approach for controlled text style transfer. In *EMNLP*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

Alexey Tikhonov, Viacheslav Shibaev, Aleksander Nagaev, Aigul Nugmanova, and Ivan P. Yamshchikov. 2019. Style transfer for texts: Retrain, report errors, compare with rewrites. In *EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*.

Cédric Villani. 2008. *Optimal transport: old and new*. Springer Science & Business Media.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a knowledge base. *arXiv preprint arXiv:1809.01797*.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational autoencoders for text generation. In *NAACL*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *EMNLP*.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *EMNLP*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2018. Learning neural templates for text generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. 2019. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *ACL*.

Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. 2018. A fast proximal point method for Wasserstein distance. In *arXiv:1802.04307*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *NeurIPS*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Ruiyi Zhang, Changyou Chen, Chunyuan Li, and Lawrence Carin. 2018. Policy optimization as wasserstein gradient flows. In *ICML*.

Ruiyi Zhang, Tong Yu, Changyou Chen, and Lawrence Carin. 2019. Text-based interactive recommendation via constraint augumented reinforcement learning. In *NeurIPS*.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *ICML*.