Keyphrase Generation via Soft and Hard Semantic Corrections

Guangzhen Zhao[†][*], **Guoshun Yin**[†][*], **Peng Yang**[†][‡], **Yu Yao**[†]
[†] School of Computer Science and Engineering, Key Laboratory of Computer Network
and Information Integration, Ministry of Education, Southeast University, China
`{zhaogz, gsyin, pengyang, yuyao2019}@seu.edu.cn`

Abstract

Keyphrase generation aims to generate a set of condensed phrases given a source document. Although maximum likelihood estimation (MLE) based keyphrase generation methods have shown impressive performance, they suffer from the bias on the source-prediction pair and the bias on the prediction-target pair. To tackle the above biases, we propose a novel correction model CorrKG on top of the MLE pipeline, where the biases are corrected via the optimal transport (OT) and a frequency-based filtering-and-sorting (FreqFS) strategy. Specifically, OT is introduced as the *soft correction* to facilitate the alignment of salient information and rectify the semantic bias on the source document and predicted keyphrases pair. An adaptive semantic mass learning scheme is conducted on the vanilla OT to achieve a proper pair-wise optimal transport procedure, which promotes the OT calculation brought by rectifying semantic masses dynamically. Besides, the FreqFS strategy is designed as the *hard correction* to reduce the bias of predicted and target keyphrases, and thus generate accurate and sufficient keyphrases. Extensive experiments over multiple benchmark datasets show that our model achieves superior keyphrase generation as compared with the state-of-the-arts.

Source Document	Learning Weights for the Quasi Weighted Means. We study the determination of weights for quasi weighted means (also called quasi linear means) when a set of examples is given. We consider first a simple case, the learning of weights for weighted means, and then we extend the approach to the more general case of a quasi weighted mean. We consider the case of a known arbitrary generator f. The paper finishes considering the use of parametric functions that are suitable when the values to aggregate are measure values or ratio.
Ground Truth	learning; quasi weighted means; quasi linear means; parametric functions; measure values; ratio values
catSeqD	weights; quasi linear means; quasi weighted mean square error
BART (Greedy Search)	learning weights; weighted means; quasi linear means; parametric functions; mean square
BART (Beam Search)	beam1: learning; weighted means; quasi linear means; ratio
	beam2: learning weights; weighted means; quasi linear means; parametric functions; mean square
	beam3: learning; weighted mean; quasi linear means; parametric functions
	beam4: learning; quasi weighted means; parametric functions; aggregate; values
	beam5: learning; quasi weighted means; quasi linear means; parametric functions; ratio
CorrKG	parametric functions; learning; quasi linear means; quasi weighted means; weighted means; ratio functions; ratio values

Figure 1: An example of keyphrase generation. The cyan reflects the bias on the source-prediction pair while the red is the bias on the prediction-target pair.

1 Introduction

Keyphrase generation is an important and meaningful task that converts the main semantic information of the document into multiple keyphrases. Keyphrases can further be divided into present keyphrases and absent keyphrases, with the former appearing in the document whereas the latter do not. High-quality keyphrases are beneficial for many downstream tasks, such as text summarization (Wang and Cardie, 2013), document clustering (Hammouda et al., 2005), translation (Tang et al., 2016), and so forth. Despite the promising success, there still exist two biases in current methods. The first one is that plenty of generated keyphrases do not match a consistent semantic description of a source document. For instance, the generated keyphrase “error” by catSeqD (Yuan et al., 2020) in Figure 1 goes beyond the content of the source document, which means there exists a bias, i.e., the semantic unfaithfulness between the source document and generated keyphrases. The underlying reason may be that Maximum Likelihood Estimation (MLE)-driven models ignore the correspondences on the source-prediction sequence pair. Meanwhile, the second bias is that there are some discrepancies between the predicted and target keyphrases. Take an example from Figure 1, the predicted keyphrase “learning weights” from BART (Lewis et al., 2020) is semantically similar to the target keyphrase “learning”, with the redundant word “weights”. A possible explanation for this phenomenon is that MLE based methods leverage greedy search by choosing the one with the maximum probability as the target, which neglects the target keyphrases that may appear in suboptimal candidates. In contrast, beam search based methods select top-k keyphrase sequences rather than the maximum one and thus are more likely to incorporate target keyphrases. However, this

[*] The first two authors contribute equally to this work.
[‡] Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7757–7768
December 7-11, 2022 ©2022 Association for Computational Linguistics

may result in a large amount of noisy keyphrases being generated in the top-k predictions. Above greedy or beam search based decoding imposes its own limited aspect, causing the second bias in keyphrase generation.

In order to alleviate the above biases, we propose a correction model on top of the MLE backbone, namely **CorrKG**, including a soft correction mechanism based on the optimal transport (OT) theory and a hard correction mechanism with a **Freq**uency-based **F**iltering and **S**orting (FreqFS) strategy. In the process of the soft correction, the OT is adopted to transform the source semantic distribution to the predicted target semantic distribution with a minimum transport cost (i.e., OT distance). Hence, the OT distance can be utilized as a correction term to measure the difference between the two semantic distributions and supervise the model to focus on salient semantic information conveying. Since not all source document tokens are equally contributed to keyphrase generation, uniform distribution may hinder proper semantic mass transport in the OT distance calculating. Therefore, to reduce the negative pair-wise assignment procedure, we equip the vanilla OT with cross-attention weights to adaptively adjust the semantic distributions, instead of directly adopting the equal masses generated from the uniform distribution like previous works (Chan et al., 2019; Kusner et al., 2015). In addition, we introduce BERT-score to evaluate semantic consistency in the soft correction procedure. Furthermore, the FreqFS strategy is designed to rectify the bias between the predicted and target keyphrases. As shown in Figure 1, it can be found that the more frequently a keyphrase appears, the more possible it is to be a ground truth (e.g., "learning", "quasi linear means"). Inspired by the above observation, the FreqFS strategy first explores multiple keyphrase sequences through beam search. Then the FreqFS strategy filters out keyphrases depending on their frequencies and subsequently sorts preserved keyphrases. In doing so, the FreqFS strategy can act as a hard correction mechanism to reduce the bias between the predicted and target keyphrases. The main contributions are listed as below:

(1) Building upon the MLE optimized BART backbone, we propose to correct the supervised model by the soft correction mechanism based on optimal transport technique and the hard correction mechanism with the FreqFS strategy.

(2) We extend the vanilla OT technique with an adaptive mass learning scheme that is capable of rectifying semantic masses dynamically, and introduce BERT-score to quantitatively evaluate the gains of semantic consistency brought by the improved OT.

(3) Extensive experiments on several popular benchmarks show that CorrKG outperforms state-of-the-art solutions and the results verify the effectiveness of the proposed model.

2 Related Work

2.1 Keyphrase Extraction and Generation

Existing methods for keyphrase prediction can be mainly categorized into extraction and generation methods. The extraction methods concentrate on selecting important words or phrases from a document as keyphrases (Florescu and Caragea, 2017; Alzaidy et al., 2019; Prasad and Kan, 2019). However, keyphrase extraction methods can only predict present keyphrases.

In order to predict both present and absent keyphrases, CopyRNN (Meng et al., 2017) is the first to employ the attention-based sequence-to-sequence (Seq2Seq) model to generate keyphrases with copy mechanism (Gu et al., 2016). Then a wide range of extensions of CopyRNN follows (Chen et al., 2018, 2019). All of the above generative models are under One2One paradigm and rely on beam search to select fixed top-k candidates as the final keyphrases. However, it is unreasonable to keep fixed top-k, since different documents have different numbers of keyphrases. Yuan et al. (2020) proposes the One2Seq paradigm and applies orthogonal regularization, target encoding strategies and semantic coverage mechanism to generate diverse numbers of keyphrases. ExHiRD (Chen et al., 2020) employs a complex exclusive hierarchical decoding framework to generate keyphrases. Besides, under One2Seq paradigm, some methods (Yuan et al., 2020; Meng et al., 2021) try to use beam search to over-generate keyphrases. The most advanced paradigm is One2Set (Ye et al., 2021) and it employs additional control codes for keyphrase generation. However, these models ignore the dense semantic correspondence between the source and predictions as well as the discrepancies between the predicted and target keyphrases.

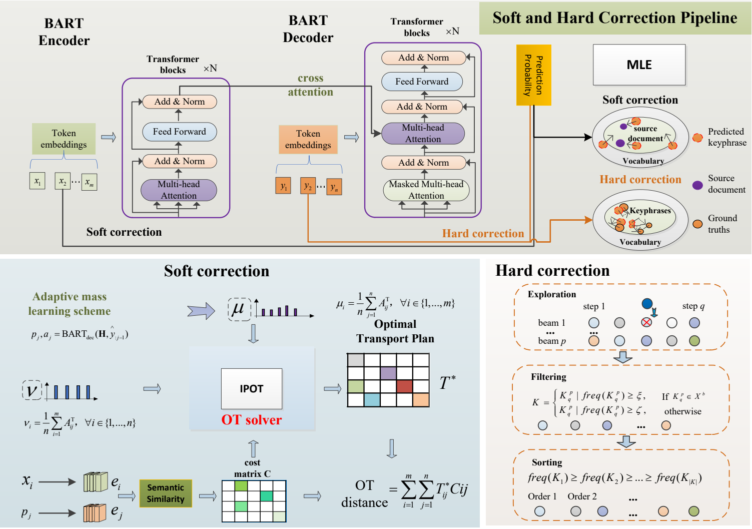


Figure 2: The overview of the CorrKG (Top). Soft correction on the source-prediction pair (Left-Bottom) and hard correction on the prediction-target pair (Right-Bottom).

2.2 Optimal Transport

Recently, optimal transport is proved to be beneficial for various NLP tasks, including document distance calculation (Kusner et al., 2015), topic modeling (Xu et al., 2018), and table-to-text generation (Wang et al., 2020). Kusner et al. (2015) proposes WMD distance to measure the dissimilarity between two documents via optimal transport. Xu et al. (2018) proposes a novel optimal-based method with a distillation mechanism for Wasserstein topic modeling. Wang et al. (2020) employs OT to establish the content-matching constraints between the text and table. To our best knowledge, we are the first to introduce the optimal transport into keyphrase generation.

3 Methodology

The proposed CorrKG consists of three components: (1) a BART backbone; (2) the soft correction mechanism; (3) the hard correction mechanism. Figure 2 shows the architecture. We details these components in the following subsections.

3.1 Problem Formulation

Given a source document $\mathbf{X} = (x_1, x_2, ..., x_{l_x})$ with l_x words, the task of keyphrase generation aims to generate a set of keyphrases $\mathcal{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^{|\mathcal{Y}|}\}$, where $|\mathcal{Y}|$ is the number of the keyphrases. Each keyphrase $\mathbf{y}^i = \left(y_1^i, \ldots, y_{l_{\mathbf{y}^i}}^i\right)$ is a sequence of keywords, where $l_{\mathbf{y}^i}$ denotes the number of words in $\mathbf{y}^i$.

3.2 Backbone

BART (Lewis et al., 2020) is a transformer-based (Vaswani et al., 2017) sequence-to-sequence pretrained model and has obtained numerous advanced results on various generative tasks. Therefore, we utilize BART as our backbone to generate keyphrases in an end-to-end manner.

We first split the source document $\mathbf{X}$ and target keyphrases $\mathbf{Y}$ using the BART tokenizer (Lewis et al., 2020). Each $\mathbf{X}$ and $\mathbf{Y}$ are tokenized into two sequences of tokens $\mathbf{X}^b = \{x_1^b, x_2^b, ..., x_m^b\}$ and $\mathbf{Y}^b = \{y_1^b, y_2^b, ..., y_n^b\}$ respectively, where m and n are the numbers of tokens, x_1^b and y_1^b indicate the start special token <s>, and x_m^b and y_n^b are the end token </s>. Then the BART encoder transforms $\mathbf{X}^b$ to $\mathbf{H} = \text{BART}_{\text{enc}}(\mathbf{X}^b) \in \mathbb{R}^{m \times d}$, where d is the hidden dimension. After obtaining the source document representation $\mathbf{H}$ and previous decoder outputs $\hat{y}_{:t-1}^b$, the BART decoder generates the probability distribution $p_t \in \mathbb{R}^{|\mathcal{V}|}$ over the predefined vocabulary $\mathcal{V}$[1], the cross-attention weights $a_t = \{a_t^1, ..., a_t^Z\}$ and the t-th output token $\hat{y}_t^b = \underset{|\mathcal{V}|}{\arg\max}(p_t)$:

$$p_t, a_t = \text{BART}_{\text{dec}}(\mathbf{H}, \hat{y}_{:t-1}^b), \tag{1}$$

where Z is the number of heads. The widely used Maximum Likelihood Estimation loss $\mathcal{L}_{\text{MLE}}$ is adopted to train the backbone model:

$$\mathcal{L}_{\text{MLE}} = -\sum_{t=1}^{n} \log p_\theta \left(y_t^b \mid \boldsymbol{y}_{:t-1}^b, \mathbf{X}^b\right). \tag{2}$$

[1] $\mathcal{V}$ contains special tokens <s> and </s>.

3.3 Soft and Hard Correction Mechanisms

3.3.1 Soft Correction Mechanism based on Optimal Transport

Optimal Transport To correct the semantic inconsistency between the source document and generated keyphrases, we formulate the keyphrase generation as an optimal transport problem. Given a source distribution μ and the corresponding target distribution ν, optimal transport attempts to gain minimal transportation cost between the both. In particular, μ and ν are sampled from probability space $\mathbb{X}, \mathbb{Y} \in \Omega$, respectively. When the cost function $c(x, y) : \mathbb{X} \times \mathbb{Y} \mapsto \mathbb{R}^{+}$ is constructed, Kantorovich (2006) addresses optimal transport problem with a probabilistic coupling $\pi \in \mathcal{P}(\mathbb{X} \times \mathbb{Y})$:

$$\boldsymbol{\pi}^{*} = \underset{\boldsymbol{\pi} \in \Pi(\mu,\nu)}{\arg\min} \int_{\mathbb{X}\times\mathbb{Y}} c(\boldsymbol{x}, \boldsymbol{y}) \boldsymbol{\pi}(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}, \quad (3)$$

where $\boldsymbol{\pi}$ is the joint probability measure given margins μ and ν, $\Pi(\mu, \nu) = \{\int_{\mathbb{Y}} \boldsymbol{\pi}(x, y) d\boldsymbol{y} = \mu, \int_{\mathbb{X}} \boldsymbol{\pi}(x, y) d\boldsymbol{x} = \nu, \boldsymbol{\pi} \geq \mathbf{0}\}$. In this paper, since the OT is applied on the textual data, we only utilize OT between the discrete source distribution $\mu = \sum_{i=1}^{m} p_i^s \delta(x_i)$ and the discrete target distribution $\nu = \sum_{j=1}^{n} p_j^t \delta(y_j)$. $\delta(x_i)$ is the Dirac function centered on x_i, m and n are the number of samples. p_i^s and p_j^t denote the corresponding probability mass, which respectively belong to the m and n-dimensional probability simplex, i.e., $\sum_{i=1}^{m} p_i^s = \sum_{i=1}^{n} p_j^t = 1$. A cost matrix $\mathbf{C}$ is defined with $\mathbf{C}_{ij}$ denoting the transport cost from sample i to sample j. Under such a setting, the optimal transport problem can be formulated as:

$$\begin{aligned} T^{*} = &\underset{T \in \mathbb{R}_{+}^{m\times n}}{\arg\min} \sum_{ij} T_{ij} \mathbf{C}_{ij} \\ &\text{s.t. } T\mathbf{1}_n = \mu, T^{\top}\mathbf{1}_m = \nu, \end{aligned} \quad (4)$$

where T^{*} is the transport matrix or optimal transport plan so as to gain an overall minimum cost $\sum_{ij} T_{ij} \mathbf{C}_{ij}$, i.e., the OT distance. T_{ij} is the mass transported from x_i to y_j.

However, it is intractable to compute the exact T^{*} (Arjovsky et al., 2017; Salimans et al., 2018). Hence the recently proposed Inexact Proximal point method for Optimal Transport (IPOT)[2] (Xie et al., 2019) is adopted to approximate the optimal transport plan and OT distance.

Adaptive Mass Learning Scheme When the OT distance is introduced into the keyphrase generation, we denote μ and ν as the semantic distribution of source document $\mathbf{X}^b$ and predicted keyphrases $\hat{\boldsymbol{Y}}^b$, respectively. The OT distance can be regarded as a metric to evaluate the strength of semantic correlation between $\mathbf{X}^b$ and $\hat{\boldsymbol{Y}}^b$. If there is no any prior knowledge, μ and ν are usually set to uniform distributions (Kusner et al., 2015; Chan et al., 2019), indicating the same importance of each token in the source document. However, assigning the same mass to each token in the source document is unreasonable in keyphrase generation, since it ignores the apparent semantic discrepancy of different tokens and aggravates the semantic bias on the source-prediction pair.

To adaptively compute the mass of semantic information within each token between the source document and generated keyphrases, we introduce an importance-aware semantic matrix A, which can be obtained by gathering all steps' weight vectors of cross-attention between the encoder and last decoder layer $A = \{\bar{a}_1, ..., \bar{a}_n\} \in \mathbb{R}^{n\times m}$. Note that $\bar{a}_i = \frac{1}{Z} \sum_{z=1}^{Z} a_i^z$, $\sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij} = n$, and a_i^z is obtained from Eq.1. For simplicity, we denote $A^{\top} \in \mathbb{R}^{m\times n}$ as the transpose of A. Then, we can obtain the semantic distributions μ and ν via the importance-aware matrix A:

$$\mu_i = \frac{1}{n} \sum_{j=1}^{n} A_{ij}^{\top}, \forall i \in \{1, ..., m\}, \quad (5)$$

$$\nu_j = \frac{1}{n} \sum_{i=1}^{m} A_{ij}^{\top}, \forall j \in \{1, ..., n\}. \quad (6)$$

Note that the semantic distribution μ of source token sequence $\mathbf{X}^b$ is adaptively updated in training. In contrary, the semantic distribution ν of generated keyphrase token sequence $\hat{\boldsymbol{Y}}^b$ is fixed as $\{\frac{1}{n}, ..., \frac{1}{n}\}$, since each keyphrase token is equally important. We treat the solution of replacing the uniform distribution with learnable cross-attention weights as the *adaptive mass learning scheme*, which results in more reasonable calculation of OT distance.

Transport Cost between Tokens We use semantic similarity to measure the unit transport cost between the source document and predicted keyphrases. Intuitively, the higher the semantic similarity between the two tokens is, the lower the unit transportation cost between them becomes. We

[2] Details of the IPOT are described in the Appendix A.1.

adopt cosine similarity as the semantic similarity:

$$Sim(e_i, e_j) = \frac{{e_i}^\top e_j}{\|e_i\|_2 \|e_j\|_2}, \tag{7}$$

where e_i denotes the embedding vector of the source document token x_i^b, $e_j = E^\top p_j$ denotes the mean token embedding vectors of the j-th predicted keyphrase token[3], $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ indicates the token embedding matrix, and $p_j \in \mathbb{R}^{|\mathcal{V}| \times 1}$ is the probability distribution of the j-th output shown in Eq.1. Finally, we can yield the transport cost matrix $\mathbf{C} = \{c_{ij} | i = 1, ...m; j = 1, ..., n\}$ as Eq.8:

$$c_{ij} = 1 - Sim(e_i, e_j). \tag{8}$$

With the obtained OT distance, a soft correction loss is defined to correct the semantic bias between the source document and its generated keyphrases:

$$\mathcal{L}_{\text{OT}} = OT(\mu, \nu) = \sum_{i=1}^{m} \sum_{j=1}^{n} T_{ij}^* C_{ij}. \tag{9}$$

The overall training objective can be optimized as follows:

$$\mathcal{L} = \mathcal{L}_{\text{MLE}} + \lambda \mathcal{L}_{\text{OT}}, \tag{10}$$

where the hyper-parameter λ balances the losses in the objective. $\mathcal{L}_{\text{MLE}}$ and $\mathcal{L}_{\text{OT}}$ complement each other to achieve better keyphrase generation.

3.3.2 Hard Correction Mechanism with Frequency-based Filtering and Sorting Strategy

To release the bias between the noisy predicted and target keyphrases, we propose a novel **Freq**uency-based **F**iltering and **S**orting (FreqFS) strategy to generate accurate keyphrases $\hat{\mathbf{y}} = \{\hat{y}_1^b, \hat{y}_2^b, ..., \hat{y}_{|\hat{\mathbf{y}}|}^b\}$. Specifically, FreqFS first explores multiple candidate keyphrases with beam search. Starting from the initial token[4] $\langle bos \rangle$, the proposed model predicts the subsequent Y_t relying on the following recursion for $t \in \{1, ..., n\}$:

$$\begin{aligned} &Y_0 \leftarrow \langle bos \rangle, \\ &Y_t \leftarrow \underset{Y_t' \subseteq \mathcal{B}_t, |Y_t'| = k}{\operatorname{argmax}} \log p_\theta \left(Y_t' \mid \mathbf{X}^b \right), \end{aligned} \tag{11}$$

where k indicates the beam size, $\mathcal{B}_t$ denotes the candidate beam set at step $t > 0$. Since the vast majority of duplicate keyphrases are generated consecutively in each beam, the FreqFS strategy enforces the decoder to produce no repetition of tri-grams, causing almost no duplication in each beam. The beam set $\mathcal{B}_t$ is formulated as:

$$\begin{aligned} \mathcal{B}_t = &\{\mathbf{y}_{:t-1} \circ \hat{y}_t^b \mid \hat{y}_t^b \in \mathcal{V} \text{ and } \mathbf{y}_{:t-1} \in Y_{t-1} \\ &\text{and } \mathbf{y}_{t-2} \circ \mathbf{y}_{t-1} \circ \hat{y}_t^b \notin \text{tri-gram}(\mathbf{y}_{:t-1})\}, \end{aligned} \tag{12}$$

where $|\mathcal{B}_t| \leq |\mathcal{V}| \cdot k$, $\circ$ denotes token concatenations, and tri-gram$(\mathbf{y}_{:t-1})$ means any tri-gram tokens that appeared in $\mathbf{y}_{:t-1}$ when $t > 2$. After that, the FreqFS strategy splits the generated keyphrase token sequences $\mathbf{y}$ with delimiter ";" to yield multiple keyphrases. $\mathcal{K}_q^p$ is defined as the q-th keyphrase in the p-th beam.

Different from vanilla beam search that concatenates all beam sequences or keeps the fixed number of keyphrasesas as the final output, FreqFS filters out the candidate keyphrases whose frequencies are lower than thresholds. The frequency distribution $freq(\mathcal{K}_q^p)$ is counted by the number of occurrences for each candidate keyphrase:

$$freq(\mathcal{K}_q^p) = \sum_{\{p_1, q_1 | p_1 \neq p \cap q_1 \neq q\}} \mathbb{I}(\mathcal{K}_q^p \cap \mathcal{K}_{q_1}^{p_1}) + 1, \tag{13}$$

where $\mathbb{I}(\cdot)$ indicates an indicator function. Then the cleaned keyphrase $\mathcal{K}$ can be obtained with the filter operation:

$$\mathcal{K} = \begin{cases} \mathcal{K}_q^p \mid freq(\mathcal{K}_q^p) \geq \xi, & \text{if } \mathcal{K}_q^p \in \mathbf{X}^b \\ \mathcal{K}_q^p \mid freq(\mathcal{K}_q^p) \geq \zeta, & \text{otherwise}, \end{cases} \tag{14}$$

with the threshold ξ for present keyphrases and ζ for absent keyphrases. Finally, the FreqFS strategy sorts the cleaned keyphrases to yield the final keyphrases. The details of FreqFS are described in Alg. 1.

4 Experiment Setup

4.1 Datasets

We experiment over multiple benchmark datasets, including KP20k (Meng et al., 2017), SemEval (Kim et al., 2010), NUS (Nguyen and Kan, 2007), and Inspec (Hulth, 2003). The KP20k dataset contains 509,820 training samples, 20,000 validation samples and 20,000 testing samples. Following previous works (Yuan et al., 2020; Chen et al., 2020; Ye et al., 2021), we employ the training set of KP20k to train and utilize the testing sets from the four benchmarks to gauge all the models.[5]

[3] The soft-argmax (Zhang et al., 2017) is employed to avoid the non-differentiable operation of sampling the j-th predicted output.

[4] We omit the beam search scores for brevity.

[5] Statistics of the datasets are described in the Appendix A.2.

Algorithm 1 The Frequency-based Filtering and Sorting strategy

```
Input: Source document $X$, beam size $k$, maximum generated length $n$, and scoring function $score(X, Y) = \log p_\theta(Y \mid X)$.
Output: The cleaned keyphrases $K$.
$Y_0 \leftarrow \{0, \langle bos \rangle\}$
for $t \in \{1, ..., n\}$ do
    $\mathcal{B}_t \leftarrow \emptyset$
    for $\{s, \mathbf{y}\} \in Y_{t-1}$ do
        if $\mathbf{y}.last() == \langle /s \rangle$ then
            $\mathcal{B}_t.add(\{s, \mathbf{y}\})$
            continue
        for $y_t \in \mathcal{V}$ do
            if $\mathbf{y}_{t-2} \circ \mathbf{y}_{t-1} \circ y_t \in$ tri-gram($\mathbf{y}$) then
                $s \leftarrow -inf$ // enforce no repetition of tri-grams
            else
                $s \leftarrow score(X, \mathbf{y} \circ y)$ // calculate the beam score
            $\mathcal{B}_t.add(\{s, \mathbf{y} \circ y_t\})$
    $Y_t \leftarrow \mathcal{B}_t.top(k)$ // select the top k according to their scores
Y $\leftarrow Split(Y_n)$
$M \leftarrow \emptyset$, $F \leftarrow \emptyset$
for $\mathcal{K}_q^p \in$ Y do
    if $K_q^p \in M$ then
        continue
    else
        $freq(\mathcal{K}_q^p) = \sum_{\{p_1, q_1 | p_1 \neq p \cap q_1 \neq q\}} \mathbb{I}(\mathcal{K}_q^p \cap \mathcal{K}_{q_1}^{p_1}) + 1$
        $M.add(\mathcal{K}_q^p)$
        if ($\mathcal{K}_q^p \in X$ and $freq(\mathcal{K}_q^p) \geq \xi$) or ($\mathcal{K}_q^p \notin x$ and $freq(\mathcal{K}_q^p) \geq \zeta$) then
            $F.add(\{\mathcal{K}_q^p, freq(\mathcal{K}_q^p)\})$
        else
            continue // filter the noisy keyphrases
$R \leftarrow F.sort()$ // sort the keyphrases by their frequencies
$K \leftarrow R$.join(";") // concatenate the sorted keyphrases with ";"
return $K$
```

4.2 Baselines

We compare the performance of CorrKG against the following advanced generative baselines. **catSeq** (Yuan et al., 2020) integrates copy mechanism to an attentional encoder-decoder model. **catSeqD** (Yuan et al., 2020) extends the catSeq with orthogonal regularization. **catSeqTG-** $2RF_1$(Chan et al., 2019) finetunes catSeqTG (Chen et al., 2019) with reinforcement learning, where F_1 and Recall metrics are regarded as rewards. **ExHiRD-h** (Chen et al., 2020) performs hierarchical decoding with an exclusion mechanism under the encoder-decoder framework. **SETTRANS** (Ye et al., 2021) proposes a Transformer-based model trained under the ONE2SET paradigm with copy mechanism and additional control codes. **Prompt-KG** (Wu et al., 2022) utilizes the prefix language model as the backbone to accomplish prompt-based generation[6].

4.3 Evaluation Metrics

We follow the evaluation metrics of previous works (Chen et al., 2020; Ye et al., 2021) and engage the macro-average $F_1@5$ and $F_1@M$ for evaluation. $F_1@5$ is calculated through comparing the top five predicted keyphrases with the target keyphrases. If the number of generated keyphrase is less than five, incorrect keyphrases are randomly appended until it reaches five predictions. $F_1@M$ concerns all the predicted keyphrases to globally access the validity of generation. Besides, we also consider BERTscore (Zhang et al., 2020) to evaluate semantic consistency in keyphrase generation.

4.4 Implementation Details

Our model is implemented on the top of Huggingface's transformers (Wolf et al., 2019) and based on BART-base model[7]. During training, the model is fine-tuned with the AdamW optimizer (Loshchilov and Hutter, 2019) on two 32G Tesla V100 GPUs for 20 epochs. The batch size is 24 for each GPU. The learning rate linearly warms up to 3×10^{-5} during the first 2K steps, and then decays with the cosine schedule. The hyper-parameter λ in Eq. 10 is 0.1. During testing, we set the beam size, the threshold ξ, and the threshold ζ as 50, 13, and 2. We repeat all of the experiments with three different random seeds and the average results are reported.

5 Results and Analysis

5.1 Present and Absent Keyphrase Predictions

Table 1 and Table 2 depict the results of the present and absent predictions, respectively. From them,

[6] We refer to this model as Prompt-KG.

[7] https://huggingface.co/facebook/bart-base

Model	Inspec		NUS		SemEval		KP20k	
	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$
catSeq (Yuan et al., 2020)	0.235	0.273	0.309	0.376	0.247	0.292	0.288	0.365
catSeqD (Yuan et al., 2020)	0.223	0.264	0.318	0.393	0.230	0.279	0.280	0.359
catSeqTG-$2RF_1$ (Chan et al., 2019)	0.253	0.301	0.375	0.433	0.287	0.329	0.321	0.386
ExHiRD-h (Chen et al., 2020)	0.253	0.291	-	-	0.284	0.335	0.311	0.374
SETTRANS (Ye et al., 2021)	0.285	0.324	0.406	**0.450**	0.331	0.357	0.358	0.392
Prompt-KG (Wu et al., 2022)	0.260	0.294	**0.412**	0.439	0.329	0.356	0.351	0.355
CorrKG	**0.330**$_4$	**0.365**$_6$	0.405$_6$	0.449$_6$	**0.333**$_8$	**0.359**$_3$	**0.372**$_0$	**0.404**$_1$

Table 1: Present keyphrase prediction results. The best results are bold, and the second-best baseline is underlined. The subscript represents the corresponding standard deviation (e.g., 0.330$_4$ indicates 0.330 ± 0.004)

Model	Inspec		NUS		SemEval		KP20k	
	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$
catSeq (Yuan et al., 2020)	0.003	0.004	0.020	0.036	0.015	0.021	0.015	0.032
catSeqD (Yuan et al., 2020)	0.007	0.013	0.013	0.022	0.018	0.025	0.014	0.030
catSeqTG-$2RF_1$ (Chan et al., 2019)	0.012	0.021	0.019	0.031	0.021	0.030	0.027	0.050
ExHiRD-h (Chen et al., 2020)	0.011	0.022	-	-	0.017	0.025	0.016	0.032
SETTRANS (Ye et al., 2021)	0.021	0.034	0.042	0.060	0.026	0.034	0.036	0.058
Prompt-KG (Wu et al., 2022)	0.017	0.022	0.036	0.042	0.028	0.032	0.032	0.042
CorrKG	**0.032**$_2$	**0.045**$_2$	**0.061**$_2$	**0.079**$_6$	**0.039**$_1$	**0.044**$_2$	**0.053**$_1$	**0.071**$_0$

Table 2: Absent keyphrase prediction results. The best results are bold, and the second-best baseline is underlined.

we can observe that CorrKG yields optimal performances on the majority of metrics, surpassing SETTRANS and Prompt-KG by a substantial margin. For instance, CorrKG promotes the best rank-1 score of current works on KP20k dataset, i.e., 1.4%/1.2% present $F_1@5/F_1@M$ and 1.7%/1.3% absent $F_1@5/F_1@M$ higher than the highest method SETTRANS, which demonstrates the prominent capability of CorrKG to correct the biases in the keyphrase generation task. Besides, we conduct paired t-test on the experiment results and find our method outperform all comparisons with p-value<0.01 on overall present and absent metrics. The promising performance is largely attributed to the improved optimal transport in faithful semantic correction and the FreqFS strategy in accurate generation.

Model	Inspec		NUS		SemEval		KP20k	
	#PK	#AK	#PK	#AK	#PK	#AK	#PK	#AK
Oracle	7.20	2.57	5.65	5.15	6.12	8.31	3.31	1.95
catSeq	4.22	0.77	3.69	0.97	4.08	0.99	3.69	0.69
catSeqD	3.94	0.69	3.46	0.75	3.54	0.88	**3.64**	0.58
catSeqTG-$2RF_1$	3.35	**2.84**	3.87	2.66	3.69	2.59	3.86	2.75
ExhiRD-h	4.00	1.50	-	-	3.65	0.99	3.97	0.81
SETTRANS	4.36	2.08	**4.80**	2.27	4.62	2.18	5.10	**2.01**
CorrKG	**5.57**	4.28	8.70	**3.68**	**7.42**	**3.62**	6.14	3.52

Table 3: Comparisons on the number of predicted keyphrases. #PK and #AK are the average number of unique present and absent keyphrases, respectively. Oracle is the ground truth average keyphrase number. #PK and #AK closest to the Oracle are bold.

5.2 Number of Predicted Keyphrases

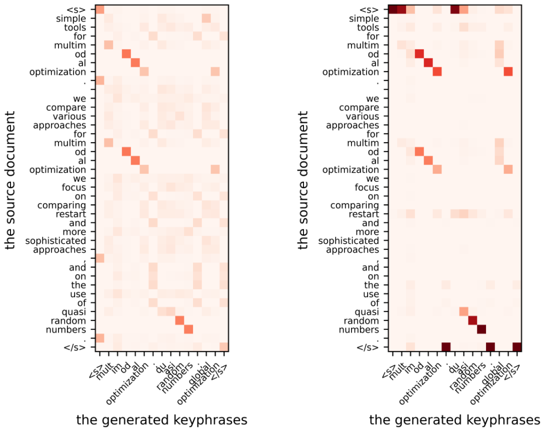
The proposed CorrKG also reveals superior diversity which can be reflected by the average numbers of unique present and absent keyphrases as illustrated in Table 3. Specifically, we can observe that accurately predicting the number of keyphrases is not trivial since none of the compared methods has an absolute advantage. However, CorrKG could outperform these baselines to some extent. We infer that, FreqFS mainly contributes to correcting the bias between the predicted and target keyphrases, which can generate more sufficient and effective keyphrases with the filtering and sorting operations. Besides, the soft correction based on OT maintains the semantic consistency between the source document and predicted keyphrases, enabling the average number of predicted keyphrases relatively stable. In summary, the OT and FreqFS cooperatively guide the model towards the direction of generating both diverse and accurate keyphrases.

5.3 Semantic Consistency Evaluation

To facilitate a more intuitive understanding of the effort of semantic correction, BERT-score is

Model	Inspec	NUS	SemEval	KP20k
catSeqTG-$2RF_1$	0.726	0.714	0.735	0.736
ExHiRD-h	0.727	-	0.701	0.725
SETTRAN	0.737	0.746	0.744	0.754
CorrKG	**0.762**	**0.759**	**0.758**	**0.766**

Table 4: Comparisons of the BERT-score on four methods. The best BERT-scores are bold.



(a) Uniform distribution mass assignment

(b) Adaptive mass learning scheme

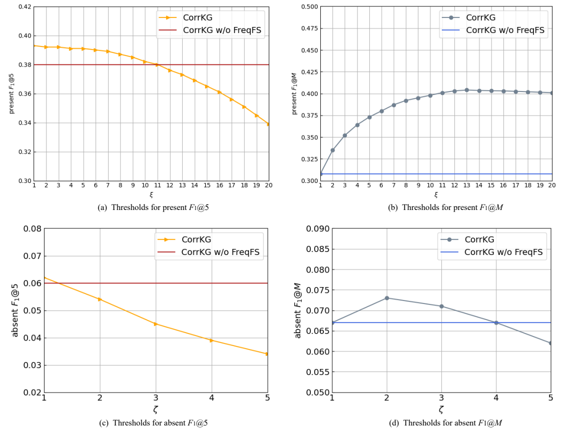
Figure 3: Semantic matching with uniform distribution and adaptive mass learning scheme.

first employed to evaluate the semantic consistency between the source document and generated keyphrases. BERT-score (Zhang et al., 2020) calculates the semantic similarity between word embeddings through BERT. As shown in Table 4, the CorrKG beats several baselines by a significant margin, indicating that the improved optimal transport solution can best fit the source document semantically. Furthermore, the consistent superior metric scores in Table 2 also agree with our motivation that the improved OT enables the generated keyphrases and source document being in the similar space. As for the hard correction between the predicted and target keyphrases, we straightforward utilize the metrics of $F_1@5$ and $F_1@M$ to access the semantic consistency. The ablation study in Table 5 sufficiently manifests the superiority of the hard correction, which will be discussed in section 5.6.2.

5.4 Optimal Transport Analysis

To further investigate how the OT contributes to the soft correction, we randomly select a training example and visualize the heat map of the optimal transport plan under the uniform distribution and the adaptive mass learning scheme settings, respectively. We have some observations from Figure 3. Compared to uniform distribution, the transport plan of the adaptive mass learning scheme has less noise, indicating its superiority in assigning importance-aware masses for OT calculating. In addition, under the regularization of OT, present keyphrases (e.g., "multimodal", "quasi random numbers") match the source document well in Figure 3 (b). As for the absent keyphrase "global optimization", the source document has not assigned reasonable weight for "global" with uniform distribution. On the contrary, "global" is focused by two "multimodal" tokens with the adaptive mass learning scheme. These suggest that the improved OT is crucial to capture pair-wise information and thus to generate keyphrases that are faithful to the source document.

5.5 Selection of Thresholds in FreqFS



(a) Thresholds for present $F_1@5$

(b) Thresholds for present $F_1@M$

(c) Thresholds for absent $F_1@5$

(d) Thresholds for absent $F_1@M$

Figure 4: Results of different ξ and ζ on the KP20k validation set.

We study the filter threshold ξ for present keyphrases and ζ for absent keyphrases in FreqFS strategy for determining the numbers of predicted present and absent keyphrases, respectively. The KP20k validation dataset is utilized to select appropriate ξ and ζ. We vary the threshold $\xi \in \{1, ..., 20\}$, $\zeta \in \{1, ..., 5\}$, and employ CorrKG w/o FreqFS as the baseline. As shown in Figure 4, for the present case, performances on $F_1@5$ are lower than that of the baseline when ξ exceeds 11. If ξ is greater than 13, both $F_1@5$ and $F_1@M$ scores begin to decrease. For absent keyphrases, $F_1@5$ and $F_1@M$ both start to degrade when ζ is greater than 2. We find that $F_1@5$ score is always lower than that of the baseline, which makes sense because the number of absent keyphrases is less than 5 when ζ is greater than 1. Hence we keep the thresholds $\xi = 13$ and $\zeta = 2$.

Model	Present		Absent	
	F_1@5	F_1@M	F_1@5	F_1@M
BART-beams	0.367	0.306	0.052	0.061
Soft correction				
CorrKG	**0.372**	**0.404**	**0.053**	**0.071**
CorrKG w/o OT	0.368	0.398	0.048	0.067
CorrKG-uniOT	0.370	0.396	0.047	0.065
Hard correction				
CorrKG	0.372	**0.404**	0.053	**0.071**
CorrKG w/o Sorting	0.368	0.404	0.051	0.071
CorrKG w/o Filtering	**0.383**	0.308	**0.060**	0.064
CorrKG w/o FreqFS	0.370	0.308	0.059	0.064

Table 5: Ablation results on KP20k. "BART-beams" is the BART model using beam search for predictions. "CorrKG-uniOT" means we replace the adaptive mass learning scheme with uniform distribution to calculate OT distance. "CorrKG w/o FreqFS" denotes the filtering and sorting operations are removed.

5.6 Ablation Study

To validate the effect of each component for CorrKG, we ablate the model and depict the results on the KP20k test dataset in Table 5.

5.6.1 Effect of OT

Ablation study on the soft correction clearly indicates the advantage of OT contribution. First, when the soft correction based on OT is removed, the decline of CorrKG w/o OT indicates that building the salient constraint between the source document and predictions is beneficial to keyphrase generation. Second, CorrKG-uniOT reveals obvious performance drops when the adaptive mass learning scheme is replaced with uniform distribution. This suggests that uniform distribution can hardly learn the optimal semantic masses. Moreover, CorrKG-uniOT performs the worst compared with CorrKG and CorrKG w/o OT, which further implies that unreasonable setting of semantic mass would hinder OT.

5.6.2 Effect of FreqFS

From Table 5, the removal of the sorting operation leads to the degradation of the F_1@5 score, showing its effectiveness. The results further confirm that the higher the frequency of a phrase, the more likely it is to be a keyphrase. It is worth noticing that the F_1@M score remains invariant whether or not the sorting operation is used, as the sorting operation does not affect the number of keyphrases. The apparent deficits of the F_1@M score for CorrKG w/o Filtering show that the filtering operation can eliminate a lot of noisy keyphrases. However, the filtering operation results in the F_1@5 score slightly decreasing, which is consistent with Figure 4 (a) and (c). The reason may be that the filtering operation would inevitably filter out a small number of correct keyphrases and make the number of predictions sometimes less than 5, resulting in the deterioration of the F_1@5 score.

6 Conclusion

Towards the semantic biases that existed in keyphrase generation, this paper presents a soft-hard correction mechanism oriented model CorrKG. The novelty in our model is twofold. First, Optimal transport theory is introduced to correct the semantic bias between the source document and predictions. Furthermore, an adaptive mass learning scheme is designed to better fit OT. Second, the FreqFS strategy is proposed to exploit the consistencies between the predicted and target keyphrases. Extensive experiments show that CorrKG is capable of generating high-quality and diverse keyphrases.

Limitations

The proposed CorrKG model has some limitations. First, the computation cost is relatively expensive due to the complicated OT calculation. Although the larger version of BART could provide more power to boost metric scores, we have to select the base version of BART in CorrKG for efficient computation. Second, we note that the soft correction based on OT works better if the semantic mass distribution is reasonably well assigned, as the adaptive mass learning scheme is more suitable than the uniform distribution in keyphrase generation. We expect this correction mechanism can be further investigated and we leave this question to future work.

Acknowledgements

The authors thank the anonymous reviewers for their valuable comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62272100, and in part by the Fundamental Research Funds for the Central Universities and the Academy-Locality Cooperation Project of Chinese Academy of Engineering under Grant JS2021ZT05.

References

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The World Wide Web Conference, WWW 2019*, pages 2551–2557, San Francisco, CA, USA.

Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, Sydney, NSW, Australia.

Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019. Neural keyphrase generation via reinforcement learning with adaptive rewards. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2163–2174, Florence, Italy.

Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase generation with correlation constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4057–4066, Brussels, Belgium.

Wang Chen, Hou Pong Chan, Piji Li, and Irwin King. 2020. Exclusive hierarchical decoding for deep keyphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1095–1105, Online.

Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R. Lyu. 2019. Title-guided encoding for keyphrase generation. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6268–6275, Honolulu, Hawaii, USA.

Corina Florescu and Cornelia Caragea. 2017. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada.

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

Khaled M. Hammouda, Diego N. Matute, and Mohamed S. Kamel. 2005. Corephrase: Keyphrase extraction for document clustering. In *Machine Learning and Data Mining in Pattern Recognition, 4th International Conference*, volume 3587, pages 265–274, Leipzig, Germany.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223, Sapporo, Japan.

Leonid V Kantorovich. 2006. On the translocation of masses. *Journal of mathematical sciences*, 133(4):1381–1382.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 957–966, Lille, France.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations*, New Orleans, LA, USA.

Rui Meng, Xingdi Yuan, Tong Wang, Sanqiang Zhao, Adam Trischler, and Daqing He. 2021. An empirical study on neural keyphrase generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4985–5007, Online.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *10th International Conference on Asian Digital Libraries*, volume 4822, pages 317–326, Hanoi, Vietnam.

Animesh Prasad and Min-Yen Kan. 2019. Glocal: Incorporating global information in local convolution for keyphrase extraction. 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 1837–1846, Minneapolis, Minnesota.

Tim Salimans, Han Zhang, Alec Radford, and Dimitris N. Metaxas. 2018. Improving gans using optimal transport. In *6th International Conference on Learning Representations*, Vancouver, BC, Canada.

Yaohua Tang, Fandong Meng, Zhengdong Lu, Hang Li, and Philip L. H. Yu. 2016. Neural machine translation with external phrase memory. *CoRR*, abs/1606.01792.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, Long Beach, CA, USA.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405, Sofia, Bulgaria.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1072–1086, Online.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Huanqin Wu, Baijiaxin Ma, Wei Liu, Tao Chen, and Dan Nie. 2022. Fast and constrained absent keyphrase generation by prompt-based learning. In *The Thirty-Sixty AAAI Conference on Artificial Intelligence*.

Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. 2019. A fast proximal point method for computing exact wasserstein distance. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 433–453, Tel Aviv, Israel.

Hongteng Xu, Wenlin Wang, Wei Liu, and Lawrence Carin. 2018. Distilled wasserstein learning for word embedding and topic modeling. In *Advances in Neural Information Processing Systems*, volume 31, Montréal, Canada.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2Set: Generating diverse keyphrases as a set. 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 4598–4608, Online.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2020. One size does not fit all: Generating and evaluating variable number of keyphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7961–7975, Online.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 4006–4015, Sydney, NSW, Australia.

A Appendix

A.1 Inexact Proximal point method for Optimal Transport(IPOT)

The details of Inexact Proximal point method for Optimal Transport (IPOT) are shown in Alg. 2.

Algorithm 2 IPOT algorithm

```
Input: Probabilities $\{\mu, \nu\}$ on support points $\{x_i\}_{i=1}^m, \{y_j\}_{j=1}^n$, cost matrix $C$ and generalized stepsize $1/\beta$
Output: $\langle T, C \rangle$
$b \leftarrow \frac{1}{m}1_m$
$G_{ij} \leftarrow e^{-\frac{C_{ij}}{\beta}}$
$T^{(1)} \leftarrow 11^\top$
for $t = 1, 2, 3, ...$ do
    $Q \leftarrow G \bigodot T^{(t)}$ // $\bigodot$ is Hadamard product
    for $l = 1, ...L$ // Usually set L=1 do
        $a \leftarrow \frac{\mu}{Qb}, b \leftarrow \frac{\nu}{Q^\top a}$
    $T^{(t+1)} \leftarrow diag(a)Qdiag(b)$
return $\langle T, C \rangle$
```

A.2 Statistics of the Testing Set

We conduct experiments over multiple public datasets, including KP20k (Meng et al., 2017), SemEval (Kim et al., 2010), NUS (Nguyen and Kan, 2007), and Inspec (Hulth, 2003). The testing dataset statistics are presented in Table 6.

Dataset	#Samples	#PK	#AK
KP20k	20,000	3.31	1.95
Inspec	500	7.20	2.57
NUS	211	5.65	5.15
SemEval	100	6.12	8.31

Table 6: Statistics of the testing set on four datasets. #PK: average number of present keyphrases. #AK: average number of absent keyphrases.