Inducing Causal Structure for Abstractive Text Summarization

Lu Chen
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
chenlu19z@ict.ac.cn

Ruqing Zhang[*]
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
zhangruqing@ict.ac.cn

Wei Huang
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
huangwei21b@ict.ac.cn

Wei Chen[†]
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
chenwei2022@ict.ac.cn

Jiafeng Guo
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
guojiafeng@ict.ac.cn

Xueqi Cheng
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
cxq@ict.ac.cn

ABSTRACT

The mainstream of data-driven abstractive summarization models tends to explore the correlations rather than the causal relationships. Among such correlations, there can be spurious ones which suffer from the language prior learned from the training corpus and therefore undermine the overall effectiveness of the learned model. To tackle this issue, we introduce a Structural Causal Model (SCM) to induce the underlying causal structure of the summarization data. We assume several latent causal factors and non-causal factors, representing the content and style of the document and summary. Theoretically, we prove that the latent factors in our SCM can be identified by fitting the observed training data under certain conditions. On the basis of this, we propose a Causality Inspired Sequence-to-Sequence model (CI-Seq2Seq) to learn the causal representations that can mimic the causal factors, guiding us to pursue causal information for summary generation. The key idea is to reformulate the Variational Auto-encoder (VAE) to fit the joint distribution of the document and summary variables from the training corpus. Experimental results on two widely used text summarization datasets demonstrate the advantages of our approach.

CCS CONCEPTS

• **Information systems** → **Summarization**; • **Computing methodologies** → *Learning latent representations.*

KEYWORDS

Abstractive text summarization, Structural causal model, VAE

ACM Reference Format:
Lu Chen, Ruqing Zhang, Wei Huang, Wei Chen, Jiafeng Guo, and Xueqi Cheng. 2023. Inducing Causal Structure for Abstractive Text Summarization. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3583780.3614934

1 INTRODUCTION

Text summarization is an important task in natural language processing (NLP), which targets to produce a fluent and condensed summary for a document, while preserving the key information [36, 45]. Abstractive summarization is a mainstream approach to generate compact summaries from scratch [3, 72]. Advances in deep learning have fueled research in applying neural sequence-to-sequence (Seq2Seq) networks to automatically extract effective features and generate summaries in an end-to-end manner [40, 47, 57].

Despite the promising performance, current data-driven summarization models possess an inherent issue. These efforts often exploit all types of correlations to fit data well, overlooking the underlying data generating process (DGP) that reveals how observed data is generated [1]. Such correlations are probably spurious due to the biased statistical dependencies caused by confounder inherited from the training corpus. For instance, if the term "lion" frequently co-occurs with "Africa" in training data, a model might erroneously generate a summary containing "Africa" even for the document describing the core information "lion pregnancy" with the side information "Africa". The occurrence of such stereotyping, arising from spurious correlations, impacts the effectiveness of text summarization techniques and hinders practical applications.

Recently, structural causal model (SCM) has attracted great interest from the research community to identify the underlying DGP of the observed data [2, 39, 49, 50, 58]. Learned causal models aid stable prediction and generalization by capturing causal relationships. In this work, we aim to devise a SCM for describing the DGP in text summarization, with the goal of inducing causal structure of the data, especially the causal relationships between documents

[*] Research conducted when the author was at the University of Amsterdam.
[†] Wei Chen is the corresponding author.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614934

and summaries. We would like the latent space to be separated into content and style space. For the content space, we assume two kinds of latent factors, i.e., Core-Content (CC) factors and Side-Content (SC) factors, referring to the core content (main points) and side content (non-essential information) in the document, respectively. For the style space, we also assume two kinds of latent factors, i.e., Document-Style (DS) factors and Summary-Style (SS) factors, referring to the lengthy writing style of the document and the concise writing style of the summary, respectively. Among such latent factors, there can be confounder representing the statistical dependencies inherited from the training corpus.

Specifically, as shown in Figure 1, we assume that CC and SS factors are summary-causal factors whose relationship with the summary remains invariant across the corpus, while other factors are non-causal for the summary and only causally influence the document. Each document is generated from the summary-causal factor CC and the non-causal factors SC and DS. Besides, we incorporate core topics and side topics in the documents to guide the learning of CC and SC factors. Theoretically, we prove that certain conditions ensure the identifiability of causal factors, enabling the generation of summaries containing only causal information.

Based on the SCM, we propose a Causality Inspired Sequence-to-Sequence model (CI-Seq2Seq) for abstractive text summarization, which enforces the learned representations to mimic the latent factors. The key idea is to learn the *causal generative mechanisms* for the document and summary, by adapting Variational Auto-encoder (VAE) [26] to supervised training. Specifically, we first partition each dataset into subsets through Latent Dirichlet Allocation (LDA) [4] and define confounder information as topical features of subsets. Then, we utilize LDA and Compression Rate (CR) [18] as the guidance to learn the content and style factors, respectively. During testing, we first infer CC and DS factors based on the learned document-causal generative mechanisms, and then use the summary-causal generative mechanisms for controlled summary generation based on the given CR between DS and SS factors.

To the best of our knowledge, it is the first work to combine causality and text summarization with a rigorous theoretical guarantee. Experimental results on two widely-used datasets, i.e., CNN/-Daily Mail [19] and XSUM [43], demonstrated that CI-Seq2Seq can achieve significant improvements over prevailing baselines in terms of prediction performance, generalizability and interpretability. The code is available at https://github.com/ict-bigdatalab/CI-Seq2Seq.

2 RELATED WORK

Text Summarization. Text summarization can be categorized into extractive and abstractive methods. Extractive methods directly extract and rearrange sentences from the document to generate the summary [40, 44, 79, 80]. Abstractive methods aim to generate a summary by comprehending the document [14, 16, 32, 47, 57]. Recently, researchers have explored to utilize pre-trained language models (PLMs) to enhance the performance of both extractive [55, 63, 69, 78] and abstractive methods [7, 28, 33, 82]. However, most studies capture only correlations rather than causal relationships.

Causality for NLP. Causality targets to explore the causal relationships in the data [50, 70], which has been widely studied in various tasks, e.g., information retrieval [24], recommendation [65, 76, 77] and computer vision [46, 60, 75]. The mainstream methods include potential outcome model [56] and structural causal model (SCM) [48, 51]. Specifically, SCM has two primary applications: causal inference and causal representation learning. The former explores variable impact via causal intervention [17] and counterfactual reasoning [5]. The latter identifies causal factors by studying data generating process, which enhances robustness and generalizability, even when facing distributional shifts [31, 35, 38, 64, 75].

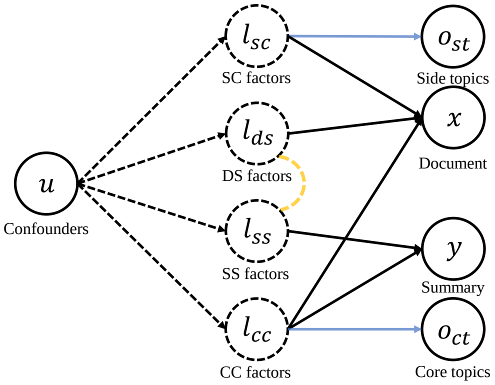

Figure 1: The proposed SCM for text summarization. Solid and dashed circles denote observed and latent variables. The solid arrows pointing to x and y represent the invariant causal generative mechanisms $p(x|l_{cc}, l_{sc}, l_{ds})$ and $p(y|l_{cc}, l_{ss})$, while the dashed arrows pointing from u represent the varied latent distributions given confounder. The blue arrows pointing to o_{st} and o_{ct} represent the content guidance for l_{sc} and l_{cc}, while the yellow dashed line between l_{ds} and l_{ss} represents their relation as the style guidance. Details see Section 3.

As for NLP, the causality-aware methods are mainly studied in text classification [53, 62], table-to-text generation [8] and language model pre-training [6], for debiasing [53], controlling [21] or style transferring [42]. For example, some works [6, 8, 21] applied causal intervention to eliminate the spurious correlations introduced by backdoor path. Some works [9, 21, 42, 53, 68] utilized counterfactual reasoning to measure the causal effect by excluding the direct effect from its total effect, or control textual attributes by assigning counterfactual values. Yet there have been few works that apply causal perspective to text summarization [68], particularly in terms of causal representation learning.

Disentangled Representation Learning. Disentangled representation learning aims to map different aspects of data into distinct low-dimensional latent spaces. It has attracted considerable attention in machine learning [20] and NLP [11, 42, 73, 81]. Besides disentangling latent factors, we focus on characterizing the causal and non-causal factors for text summarization.

3 A CAUSAL VIEW ON SUMMARIZATION

Following the definition that *two variables have a causal relationship, denoted as "cause → effect", if intervening the cause may alter the effect, but not vice versa* [50, 52], we first define the causal relationships in text summarization, and then formulate it using structural causal model (SCM) [51], followed by the identifiability analysis to ensure that the latent factors in our SCM can be correctly separated and learned under certain conditions.

3.1 Causal Relationships

We introduce step-by-step about how we characterize the causal relationships in text summarization.

(1) Assuming latent factors with causal relationships. It is likely that there exist correlations between a document x and its summary y. According to the Reichenbach's common cause principle [52], correlations mean there exist common causes that causally influence x and y. We assume latent factors z as common causes, carrying mixed information of x and y. That is, $z \rightarrow x$ and $z \rightarrow y$.

(2) Clarifying the causes for document and summary. To separate the mixed information in z, we decompose z in terms of content and style, i.e., Core-Content (CC) factor l_{cc} and Side-Content (SC) factor l_{sc} in content space, as well as Document-Style (DS) factor l_{ds} and Summary-Style (SS) factor l_{ss} in style space. For content, summary y should preserve the core content while omit the side content of document x, i.e., $l_{cc} \rightarrow y$, $l_{cc} \rightarrow x$ and $l_{sc} \rightarrow x$. For style, considering different style of x and y, we assume l_{ds} is the style factor for x and l_{ss} for y, i.e., $l_{ds} \rightarrow x$ and $l_{ss} \rightarrow y$.

(3) Capturing correlations among latent factors. Latent factors may mix through spurious correlation from biased statistical dependencies of the training corpus [41, 53]. We use u to denote confounder resulting in the spurious correlation, and orient four edges from u to latent factors, i.e., $u \rightarrow l_{cc}, u \rightarrow l_{sc}, u \rightarrow l_{ds}, u \rightarrow l_{ss}$.

(4) Adding guidance to separate latent factors. Practically, we use weakly-supervised signals to guide latent factor learning. For content factors, we introduce core topics o_{ct} and side topics o_{st} for l_{cc} and l_{sc} respectively. That is, $l_{cc} \rightarrow o_{ct}$ and $l_{sc} \rightarrow o_{st}$. For style factors, we define the function relation between l_{ds} and l_{ss} to bridge them. Thus, we link $l_{ds} - l_{ss}$ by an undirected edge.

3.2 Structural Causal Model

Based on the above analysis, we devise the SCM for text summarization (Figure 1). It describes the data generating process (DGP) – latent factors generate the observations (document and summary) given the confounder. The nodes denote variables, and the edges denote relationships (directed: causal, undirected: non-causal).

We refer to $p(x|l_{cc}, l_{sc}, l_{ds})$ and $p(y|l_{cc}, l_{ss})$ as the causal generative mechanisms for the document and summary, respectively. They are assumed to be invariant to the prior $p(l_{cc}, l_{sc}, l_{ds}, l_{ss})$ according to the Independent Causal Mechanisms (ICM) Principle [52, 59], denoted by solid arrows in Figure 1, while the latent distributions given the confounder may vary across domains, denoted by dashed arrows. Besides, the topic distributions $p(o_{ct}|l_{cc})$ and $p(o_{st}|l_{sc})$ denote the content guidance, and the function relation between l_{ds} and l_{ss} denotes the style guidance. We formally present a comprehensive functional form for the DGP as outlined below.

We define $\Theta \triangleq \{f, \Phi\}$ as the parameters to generate observed variables, where f is the invertible function mapping latent factors to observed variables, and Φ denotes the parameters to generate latent factor given confounder u. The parent set is denoted as $Pa(\cdot)$.

Taking x as an example, $Pa(x) = \{l_{sc}, l_{ds}, l_{cc}\}$. The joint probability density of x and $Pa(x)$ can be written as:

$$\begin{aligned} p_{\Theta_x}(x, Pa(x)|u) =& p_{\Theta_x}(x, l_{sc}, l_{cc}, l_{ds}|u) \\ =& p_{f_x}(x|l_{sc}, l_{cc}, l_{ds}) \cdot p_{\Phi_x}(l_{sc}, l_{cc}, l_{ds}|u). \end{aligned} \tag{1}$$

For $p_{\Phi_x}(l_{sc}, l_{cc}, l_{ds}|u)$, we assume it follows exponential family:

$$\begin{aligned} p_{\Phi_x}(l_{sc}, l_{cc}, l_{ds}|u) =& p_{\mathbf{T}^{l_{sc}}, \boldsymbol{\lambda}^{l_{sc}}}(l_{sc}|u) \cdot p_{\mathbf{T}^{l_{cc}}, \boldsymbol{\lambda}^{l_{cc}}}(l_{cc}|u) \cdot p_{\mathbf{T}^{l_{ds}}, \boldsymbol{\lambda}^{l_{ds}}}(l_{ds}|u) \\ =& \prod_{i=1}^{d_{sc}} p_{\mathbf{T}^{l_{sc}}, \boldsymbol{\lambda}^{l_{sc}}}(l_{sc_i}|u) \cdot \prod_{i=1}^{d_{cc}} p_{\mathbf{T}^{l_{cc}}, \boldsymbol{\lambda}^{l_{cc}}}(l_{cc_i}|u) \cdot \prod_{i=1}^{d_{ds}} p_{\mathbf{T}^{l_{ds}}, \boldsymbol{\lambda}^{l_{ds}}}(l_{ds_i}|u) \\ =& \prod_{Pa \in \{l_{sc}, l_{cc}, l_{ds}\}} \prod_{i=1}^{d_{Pa}} \frac{Q_i^{Pa}(Pa_i)}{Z_i^{Pa}(u)} \cdot \exp\left[\sum_{j=1}^{k_{Pa}} T_{i,j}^{Pa}(Pa_i)\lambda_{i,j}^{Pa}(u)\right], \end{aligned} \tag{2}$$

where Φ_x contains sufficient statistics $\mathbf{T}$ and coefficient $\boldsymbol{\lambda}$, Q_i^{Pa} is the base measure, and Z_i^{Pa} is the normalization function.

For $p_{f_x}(x|l_{sc}, l_{cc}, l_{ds})$, we constrain it by Additive Noise Model (ANM) assumption [23], where the DGP for x can be expressed as:

$$x = f_x(l_{sc}, l_{cc}, l_{ds}) + \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \sim p_{\boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon}). \tag{3}$$

We rewrite Equation 1 using Equation 3, i.e.,

$$p_{\Theta_x}(x, Pa(x)|u) = p_{\boldsymbol{\varepsilon}}(x - f_x(l_{sc}, l_{cc}, l_{ds})) \cdot p_{\Phi_x}(l_{sc}, l_{cc}, l_{ds}|u). \tag{4}$$

Similarly, we can obtain the results for y, o_{st} and o_{ct}, i.e.,

$$p_{\Theta_y}(y, Pa(y)|u) = p_{\boldsymbol{\varepsilon}}(y - f_y(l_{ss}, l_{cc})) \cdot p_{\Phi_y}(l_{ss}, l_{cc}|u), \tag{5}$$

$$p_{\Theta_{o_{st}}}(o_{st}, Pa(o_{st})|u) = p_{\boldsymbol{\varepsilon}}(o_{st} - f_{o_{st}}(l_{sc})) \cdot p_{\Phi_{o_{st}}}(l_{sc}|u), \tag{6}$$

$$p_{\Theta_{o_{ct}}}(o_{ct}, Pa(o_{ct})|u) = p_{\boldsymbol{\varepsilon}}(o_{ct} - f_{o_{ct}}(l_{cc})) \cdot p_{\Phi_{o_{ct}}}(l_{cc}|u). \tag{7}$$

In summary, using the DGP in our SCM, we can express the joint probability density functions as Equations 4-7.

Notice that latent variables cannot be directly obtained. Instead, we can only learn their representations by mimicking these distributions. This raises a crucial question: Can we learn representations for each latent factor without mixing information with others, while ensuring that the difference between the learned representations and the true representations remains within acceptable bounds of uncertainty? This refers to the identifiability of the latent variables.

How to ensure identifiability, i.e, how to solve the question, is presented in the subsequent section.

3.3 Identifiability Analysis

As discussed in Section 3.2, we aim to learn representations for latent factors while ensuring their identifiability. To achieve this, we begin by defining an equivalence relation denoted as $\sim_P$.

Definition 3.1 ($\sim_P$ Equivalent). Suppose Θ and $\tilde{\Theta}$ are two set of parameters for the SCM as defined in Section 3.2. Θ and $\tilde{\Theta}$ are called $\sim_P$ equivalent if the following conditions are met:

$$p_{\Theta}(o_{st}, o_{ct}, x, y) = p_{\tilde{\Theta}}(o_{st}, o_{ct}, x, y), \tag{8}$$

$$\forall o, \forall l, \exists(\mathbf{A}^l, \mathbf{c}^l), \text{ s.t. } \mathbf{T}^l([f_o]_l^{-1}(o)) = \mathbf{A}^l \tilde{\mathbf{T}}^l([\tilde{f}_o]_l^{-1}(o)) + \mathbf{v}^l \tag{9}$$

where $o \in \{x, y, o_{ct}, o_{st}\}$, l is a latent factor in $Pa(o)$, expressed as $l \in Pa(o)$, $\mathbf{A}$ is an invertible permutation matrix, and $\mathbf{v}$ is a vector.

The following Theorem 3.2 provides a sufficient condition which ensures our model to learn parameters $\tilde{\Theta}$ that satisfy $\sim_p$ equivalence with true parameters Θ.

Theorem 3.2 ($\sim_P$ Identifiability). *Considering the SCM described in Section 3.2, if we have an adequate number of distinct u values, denoted as k_u, that satisfy the variety assumption, i.e., the matrix $\mathbf{L} \triangleq [\boldsymbol{\lambda}(u_1) - \boldsymbol{\lambda}(u_0), ..., \boldsymbol{\lambda}(u_{k_u}) - \boldsymbol{\lambda}(u_0)]$ has full column rank, where $\boldsymbol{\lambda}(u)$ represents the vector parameter for the probability density function of an exponential family distribution. Then the learned parameter $\tilde{\Theta}$ and the true parameter Θ exhibit $\sim_p$ equivalent.*

Discussion Theorem 3.2 ensures that the learned parameters are $\sim_p$ equivalent with the true parameters, that is: (1) The joint distributions given by learned parameters and true parameters match

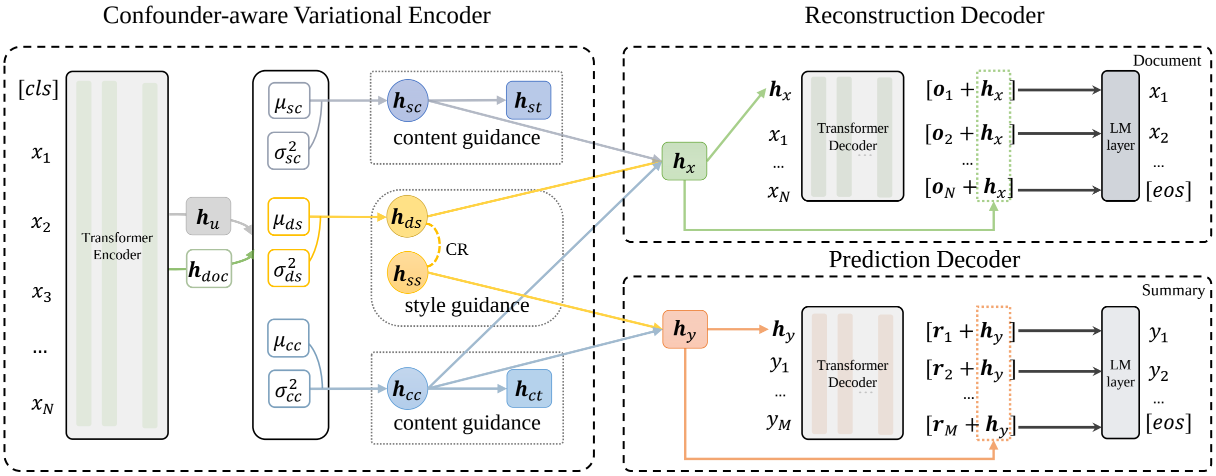

Figure 2: The overall architecture of CI-Seq2Seq model.

(Equation 8). (2) Latent factors can be separated, as only one appears in Equation 9 each time. (3) The difference between learned latent factors and true ones is limited to a permutation transformation with a linear shift applied to their sufficient statistics (Equation 9).

Besides, Theorem 3.2 requires that the number k_u of different values for confounder u is sufficient. It can be satisfied by proper definition for confounder. Proofs are provided in Section 5.

4 CAUSALITY INSPIRED SEQ2SEQ MODEL

Under the theoretical guarantees on modeling latent factors separately (Theorem 3.2), we propose the Causality Inspired Sequence-to-Sequence (CI-Seq2Seq) model to learn representations that can mimic the latent factors by fitting the observed training data.

In the following, we first present our model architecture, a restructured Variational Auto-Encoder (VAE) [26], which learns latent representations from input and produces samples resembling the original data. Then, we detail learning strategy for causal generative mechanisms $p(x|l_{cc}, l_{sc}, l_{ds})$ and $p(y|l_{cc}, l_{ss})$, followed by the controlled generation procedure using these learned mechanisms.

4.1 Model Architecture

As depicted in Figure 2, the proposed CI-Seq2Seq contains three main components: Confounder-aware Variational Encoder, Reconstruction Decoder, and Prediction Decoder.

4.1.1 *Confounder-aware Variational Encoder.*

This encoder targets to obtain representations $\mathbf{h}_{cc}$, $\mathbf{h}_{sc}$, $\mathbf{h}_{ds}$ and $\mathbf{h}_{ss}$ for the CC, SC, DS and SS factors from the input document $x = \{x_1, x_2, ..., x_N\}$ of length N. Based on Theorem 3.2, confounder u is essential in distinguishing latent factors. It can be defined as the intrinsic properties of training data, e.g., topic, style and domains. Here, we denote u as the topic extracted from documents. Then, the encoder maps x and u into $\mathbf{h}_{cc}$, $\mathbf{h}_{sc}$, $\mathbf{h}_{ds}$ and $\mathbf{h}_{ss}$ according to $q_\phi(l_{cc}, l_{sc}, l_{ds}|x, u)$ and the relation between l_{ds} and l_{ss}, where ϕ denotes the parameters of the confounder-aware variational encoder.

- **Encoding Confounder Information $\mathbf{h}_u$.** To achieve different values of confounder u, we denote u as the topical features. We first partition each summarization corpus into k_u subsets via LDA topic classification, where each document belongs to one subset. Specifically, each document obtains a topic distribution from LDA, and the topic id *tid* with the highest probability is assigned to the document. Then, following the practice of word embedding [37], *tid* is applied to look up a hidden vector $\mathbf{h}_u \in \mathbb{R}^{d_u}$ from a trainable embedding matrix $\mathbf{E}_u \in \mathbb{R}^{k_u \times d_u}$, i.e., $\mathbf{h}_u = \mathbf{E}_u(tid)$.
- **Encoding Source Information $\mathbf{h}_{doc}$.** CC, SC and DS factors are probably influenced by the full information of the document. Therefore, we propose to model the distribution of them conditioned on the global semantic representation of x. Specifically, given an input document x, we first add a special token "[CLS]" in front of it, and then leverage the final hidden state of this token as its global representation $\mathbf{h}_{doc} \in \mathbb{R}^{d_h}$. It is a flexible aggregate and comprehensive understanding of the entire sequence.
- **Sampling $\mathbf{h}_{cc}$, $\mathbf{h}_{sc}$ and $\mathbf{h}_{ds}$.** We mix $\mathbf{h}_{doc}$ with $\mathbf{h}_u$ and encode them into the distribution of $\mathbf{h}_{cc}$, $\mathbf{h}_{sc}$ and $\mathbf{h}_{ds}$ to model the posterior distributions $p(l_{cc}, l_{sc}, l_{ds}|x, u)$. Specifically, the true posterior $p(l_{cc}, l_{sc}, l_{ds}|x, u)$ is approximated via the variational distribution $q_\phi(l_{cc}, l_{sc}, l_{ds}|x, u)$. We constraint the prior distributions $p(l_{cc}, l_{sc}, l_{ds})$ as standard Gaussian distributions following [26, 29]. Gaussian parameters mean $\mu_{cc,sc,ds}$ and variance $\sigma^2_{cc,sc,ds}$ are projected from the concatenation of $\mathbf{h}_{doc}$ and $\mathbf{h}_u$:
$$[\mu_{cc};\mu_{sc};\mu_{ds}] = \mathbf{W}_1[\mathbf{h}_{doc};\mathbf{h}_u] + \mathbf{b}_1, \tag{10}$$
$$([\log\sigma^2_{cc};\log\sigma^2_{sc};\log\sigma^2_{ds}]) = \mathbf{W}_2[\mathbf{h}_{doc};\mathbf{h}_u] + \mathbf{b}_2,$$
where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{(d_h+d_u)\times(d_{cc}+d_{sc}+d_{ds})}$, $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{d_{cc}+d_{sc}+d_{ds}}$ are learnable. Finally, we sample $\mathbf{h}_{cc} \in \mathbb{R}^{d_{cc}}$, $\mathbf{h}_{sc} \in \mathbb{R}^{d_{sc}}$ and $\mathbf{h}_{ds} \in \mathbb{R}^{d_{ds}}$ from the learned distribution using a reparametrization trick [26], respectively.
- **Computing $\mathbf{h}_{ss}$ with Style Guidance.** Since the SS factors are only causally related to the summary $y = \{y_1, y_2, ..., y_M\}$ of length M, it is not suitable to directly extract them from x like DS factors. Therefore, we introduce compression rate (CR) [18] between DS and SS factors as the style guidance. Specifically, CR help bridge DS and SS factors smoothly, which indicates the information ratio between the target summary and the source

document. Following previous work [71], we define CR as the ratio of the text length between the summary and the document, i.e., $CR = M/N \in (0, 1)$. Based on CR, we can obtain $\mathbf{h}_{ss}$:

$$\mathbf{h}_{ss} = \mathbf{h}_{ds} \times CR \tag{11}$$

4.1.2 Reconstruction Decoder. This decoder targets to utilize the representations $\mathbf{h}_{cc}$, $\mathbf{h}_{sc}$ and $\mathbf{h}_{ds}$ of the CC, SC and DS factors to reconstruct the input document x according to $p_\theta(x|l_{cc}, l_{sc}, l_{ds})$, where θ denotes the parameters of the reconstruction decoder.

First, we apply a fully connected (FC) layer to combine $\mathbf{h}_{cc}$, $\mathbf{h}_{sc}$ and $\mathbf{h}_{ds}$ into the composed information $\mathbf{h}_x$. Then, we propose to replace the first token of decoder input with $\mathbf{h}_x$, since the first token matters much for the generation of following tokens. Besides, the first token is only allowed to attend to itself, which could alleviate the vanishing latent factor problem to some extent [66].

To further enhance the impact of $\mathbf{h}_x$, we add it to all the output hidden states $\{\mathbf{o}_i\}_{i=1}^{N}$ from the last Transformer layer in the reconstruction decoder. The vocabulary selection probability P_x for generating x is computed as

$$P_x = \mathbf{W}_3(\mathbf{o}_i + \mathbf{h}_x) + \mathbf{b}_3, \tag{12}$$

where $\mathbf{W}_3 \in \mathbb{R}^{d_h \times d_v}$ and $\mathbf{b}_3 \in \mathbb{R}^{d_v}$ are learnable.

4.1.3 Prediction Decoder. This decoder only allows the injection of the CC representation $\mathbf{h}_{cc}$ along with the SS representation $\mathbf{h}_{ss}$ for generating the summary y according to $p_\tau(y|l_{cc}, l_{ss})$, where τ denotes the parameters of the prediction decoder.

First, similar to the reconstruction decoder, we obtain the composed representation $\mathbf{h}_y$ for summary prediction, by combining $\mathbf{h}_{cc}$ and $\mathbf{h}_{ss}$ using a FC layer. Then, we replace the first token with $\mathbf{h}_y$ in the prediction decoder. Simultaneously, we add $\mathbf{h}_y$ to all the output hidden states $\{\mathbf{r}_j\}_{j=1}^{M}$ from the last transformer layer in the prediction decoder. The final vocabulary selection probability P_y for generating y is calculated in the same way as P_x.

4.2 Learning Strategy

To learn $p(x|l_{cc}, l_{sc}, l_{ds})$ and $p(y|l_{cc}, l_{ss})$ for invariant prediction, we reformulate the learning objective function of VAE in the supervised scenario to fit the training corpus. Specifically, we apply four learning objectives as follows.

- **Reconstruction Loss** is applied to train the reconstruction decoder to reconstruct the input document, i.e.,
$$\mathcal{L}_R = -\mathbb{E}_{q_\phi(l_{cc}, l_{sc}, l_{ds}|x,u)}[\log p_\theta(x|l_{cc}, l_{sc}, l_{ds})]. \tag{13}$$
- **Prediction Loss** is applied to encourage the prediction decoder to generate the summary based on the summary-causal representations, i.e.,
$$\mathcal{L}_P = -\mathbb{E}_{q_\phi(l_{cc}, l_{sc}, l_{ds}|x,u)}[\log p_\tau(y|l_{cc}, l_{ss})]. \tag{14}$$
- **KL Loss** is a regularizer based on the Kullback-Leibler (KL) divergence, applied to push the posterior $q_\phi(l_{cc}, l_{sc}, l_{ds}|x, u)$ to be closed to the prior $p(l_{cc}, l_{sc}, l_{ds})$ which is constrained as standard Gaussian distributions, i.e.,
$$\mathcal{L}_{KL} = \mathbb{D}_{KL}[q_\phi(l_{cc}, l_{sc}, l_{ds}|x, u) \| p(l_{cc}, l_{sc}, l_{ds})]. \tag{15}$$
- **Content Guidance Loss** is further applied to guide the optimization of the CC and SC factors, which is calculated by three steps. (i) We first extract the core topics o_{ct} and side topics o_{st} in x according to the LDA topic distribution $p(o_t|x)$ on k_u topics. Specifically, given a threshold th, a topic $o_t^a (a \in \{1, 2, ..., k_u\})$ belongs to the core topics of document x if $p(o_t = o_t^a|x) > th$, otherwise side ones. To indicate the type of each topic, we introduce a k_u−dimension binary indicator $\mathbf{g}$, where "1" represents the core topics and "0" represents the side ones. (ii) We then transform such topic information into hidden representations $\mathbf{h}_{ct}, \mathbf{h}_{st} \in \mathbb{R}^{d_h}$ based on another learnable embedding matrix $\mathbf{E}_t \in \mathbb{R}^{k_u \times d_h}$. Similar to $\mathbf{E}_u$, each row of $\mathbf{E}_t$ represents a topic embedding. Specifically, to achieve the aggregated hidden representation $\mathbf{h}_{ct}$ which combines information of all core topics, we obtain the core topic distribution $p(o_{ct}|x)$ based on the binary indicator $\mathbf{g}$, i.e.,
$$p(o_{ct}|x) = Norm(p(o_t|x) \odot \mathbf{g}), \tag{16}$$
where $\odot$ denotes element-wise multiplication and $Norm()$ denotes normalization operation. After that, we linearly combine topic embeddings from $\mathbf{E}_t$ according to $p(o_{ct}|x)$ as below:
$$\mathbf{h}_{ct} = p(o_{ct}|x)\mathbf{E}_t. \tag{17}$$
Similarly, for side topics, we have
$$p(o_{st}|x) = Norm(p(t|x) \odot (\mathbf{1} - \mathbf{g})), \tag{18}$$
$$\mathbf{h}_{st} = p(o_{st}|x)\mathbf{E}_t. \tag{19}$$
(iii) Finally, we compute the Euclidean distance (i.e., L2 distance) of $\langle \mathbf{h}_{cc}, \mathbf{h}_{ct} \rangle$ and $\langle \mathbf{h}_{sc}, \mathbf{h}_{st} \rangle$ as the content guidance loss, i.e.,
$$\mathcal{L}_{LDA} = L2(\mathbf{h}_{cc}, \mathbf{h}_{ct}) + L2(\mathbf{h}_{sc}, \mathbf{h}_{st}). \tag{20}$$

The total loss is a summation of the four losses:

$$\mathcal{L} = \mathcal{L}_R + \mathcal{L}_P + \lambda_{kl}\mathcal{L}_{KL} + \lambda_{lda}\mathcal{L}_{LDA}, \tag{21}$$

where λ_{kl} and λ_{lda} are used to control the strength of the regularization and the content guidance.

4.3 Controlled Generation

During the test stage, we first optimize the following log-likelihood to infer l_{cc}^*, l_{sc}^* and l_{ds}^*, i.e.,

$$\max_{l_{cc}, l_{sc}, l_{ds}} \log p_\theta(x|l_{cc}, l_{sc}, l_{ds}) + \lambda_{cc}\|l_{cc}\|_2^2 + \lambda_{sc}\|l_{sc}\|_2^2 + \lambda_{ds}\|l_{ds}\|_2^2, \tag{22}$$

where λ_{cc}, λ_{sc} and λ_{ds} control the learned l_{cc}, l_{sc} and l_{ds} in a reasonable scale. Specifically, we sample some candidate points from $N(0, I)$ and select the optimal one in terms of Equation 22 as the initial point for further optimization.

Finally, we employ the optimized l_{cc}^* and l_{ds}^* to generate summaries with different styles by varying CR[1]. In this way, we can actively control the compression rate of the summary. That is, with different l_{ss}^* values and the optimized l_{cc}^*, we generate the summary y based on the learned $p_\tau(y|l_{cc}^*, l_{ss}^*)$.

5 PROOF

Proof. For Theorem 3.2, we will demonstrate that we can learn a parameter $\tilde{\Theta}$ that is $\sim_P$ equivalent to the true parameter Θ, satisfying two conditions: Equation 8 and Equation 9 in Definition 3.1. The first condition means the correct fitting of the joint distribution of observed variables, which can be guaranteed by the universal approximation ability of neural networks. Therefore, our main task is to prove the validity of the second condition.

The proof is roughly divided into two steps: *Denoising* and *Identifying*. We will present the proof step by step.

(1) Denoising. This step serves the purpose of eliminating noise variables while retaining only the latent factors. Assuming that the

[1] Note that CR is the ground-truth summary-document length ratio during training.

learned distribution of observed variables is identical to the true distribution (i.e., the first condition holds), we have $p_{\Theta}(o_{st}, o_{ct}, x, y) = p_{\tilde{\Theta}}(o_{st}, o_{ct}, x, y)$. By integrating variables o_{st}, o_{ct}, and y, we can obtain $p_{\Theta_x}(x) = p_{\tilde{\Theta}_x}(x)$. Then, we express it given confounder:

$$p_{f_x, \mathbf{T}^{l_{sc}}, \boldsymbol{\lambda}^{l_{sc}}, \mathbf{T}^{l_{ds}}, \boldsymbol{\lambda}^{l_{ds}}, \mathbf{T}^{l_{cc}}, \boldsymbol{\lambda}^{l_{cc}}}(x|u) = p_{\tilde{f}_x, \tilde{\mathbf{T}}^{l_{sc}}, \tilde{\boldsymbol{\lambda}}^{l_{sc}}, \tilde{\mathbf{T}}^{l_{ds}}, \tilde{\boldsymbol{\lambda}}^{l_{ds}}, \tilde{\mathbf{T}}^{l_{cc}}, \tilde{\boldsymbol{\lambda}}^{l_{cc}}}(x|u), \tag{23}$$

$$\Rightarrow \int p_{\boldsymbol{\varepsilon}}(x - f_x(l_{sc}, l_{cc}, l_{ds}))p_{\boldsymbol{\Phi}_x}(l_{sc}, l_{cc}, l_{ds}|u) \cdot dl_{sc}dl_{cc}dl_{ds}$$

$$= \int p_{\boldsymbol{\varepsilon}}(x - \tilde{f}_x(l_{sc}, l_{cc}, l_{ds}))p_{\tilde{\boldsymbol{\Phi}}_x}(l_{sc}, l_{cc}, l_{ds}|u) \cdot dl_{sc}dl_{cc}dl_{ds}, \tag{24}$$

$$\Rightarrow \int p_{\boldsymbol{\varepsilon}}(x - \bar{x})p_{\boldsymbol{\Phi}_x}(f_x^{-1}(\bar{x})|u) \left|\det(J_{f_x^{-1}}(\bar{x}))\right| d\bar{x}$$

$$= \int p_{\boldsymbol{\varepsilon}}(x - \tilde{x})p_{\tilde{\boldsymbol{\Phi}}_x}(\tilde{f}_x^{-1}(\tilde{x})|u) \left|\det(J_{\tilde{f}_x^{-1}}(\tilde{x})\right| d\tilde{x}, \tag{25}$$

$$\Rightarrow \int p_{\boldsymbol{\varepsilon}}(x - \bar{x})p_{\boldsymbol{\Phi}_x, f_x, u}(\bar{x})d\bar{x} = \int p_{\boldsymbol{\varepsilon}}(x - \tilde{x})p_{\tilde{\boldsymbol{\Phi}}_x, \tilde{f}_x, u}(\tilde{x})d\tilde{x}, \tag{26}$$

$$\Rightarrow \quad (p_{\boldsymbol{\varepsilon}} * p_{\boldsymbol{\Phi}_x, f_x, u})(x) = (p_{\boldsymbol{\varepsilon}} * p_{\tilde{\boldsymbol{\Phi}}_x, \tilde{f}_x, u})(x), \tag{27}$$

$$\Rightarrow \quad F[p_{\boldsymbol{\varepsilon}}](\omega)F[p_{\boldsymbol{\Phi}_x, f_x, u}](\omega) = F[p_{\boldsymbol{\varepsilon}}](\omega)F[p_{\tilde{\boldsymbol{\Phi}}_x, \tilde{f}_x, u}](\omega), \tag{28}$$

$$\Rightarrow \quad p_{\boldsymbol{\Phi}_x, f_x, u}(x) = p_{\tilde{\boldsymbol{\Phi}}_x, \tilde{f}_x, u}(x). \tag{29}$$

In Equation 25, J represents the Jacobian matrix, and $|\det|$ denotes generalized determinant, defined as $|\det(A)| \triangleq \sqrt{\det(A^\top A)}$. The symbol $\triangleq$ is read as "is defined as". Equation 26 introduces the function $p_{\boldsymbol{\Phi}x, f, u}(\bar{x}) \triangleq p_{\boldsymbol{\Phi}x}(f_x^{-1}(\bar{x})|u) \det(J_{f_x^{-1}}(\bar{x}))$ for convenience. Note that Equation 26 corresponds to a convolution operation as expressed in Equation 27. In Equation 28, F means Fourier transformation, a useful tool to simplify convolution. In Equation 29, we obtain the denoised result. Similar results can be obtained for the other observed variables y, o_{st}, and o_{ct}, i.e.,

$$p_{\boldsymbol{\Phi}_y, f_y, u}(y) = p_{\tilde{\boldsymbol{\Phi}}_y, \tilde{f}_y, u}(y), \tag{30}$$

$$p_{\boldsymbol{\Phi}_{o_{st}}, f_{o_{st}}, u}(o_{st}) = p_{\tilde{\boldsymbol{\Phi}}_{o_{st}}, \tilde{f}_{o_{st}}, u}(o_{st}), \tag{31}$$

$$p_{\boldsymbol{\Phi}_{o_{ct}}, f_{o_{ct}}, u}(o_{ct}) = p_{\tilde{\boldsymbol{\Phi}}_{o_{ct}}, \tilde{f}_{o_{ct}}, u}(o_{ct}). \tag{32}$$

Furthermore, notice that different observed variables share common latent factors. To capture this characteristic, we specifically target pairs of observed variables and apply the aforementioned denoising method to these pairs. This idea is inspired by LaCIM [60]. For the variable pairs (x, o_{st}), (x, o_{ct}) and (y, o_{ct}), we can obtain the similar denoised results:

$$p_{\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_{o_{st}}, f_a, u}(a) = p_{\tilde{\boldsymbol{\Phi}}_x, \tilde{\boldsymbol{\Phi}}_{o_{st}}, \tilde{f}_a, u}(a), \tag{33}$$

$$p_{\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_{o_{ct}}, f_b, u}(b) = p_{\tilde{\boldsymbol{\Phi}}_x, \tilde{\boldsymbol{\Phi}}_{o_{ct}}, \tilde{f}_b, u}(b), \tag{34}$$

$$p_{\boldsymbol{\Phi}_y, \boldsymbol{\Phi}_{o_{ct}}, f_c, u}(c) = p_{\tilde{\boldsymbol{\Phi}}_y, \tilde{\boldsymbol{\Phi}}_{o_{ct}}, \tilde{f}_c, u}(c), \tag{35}$$

where $a \triangleq [x^\top, o_{st}^\top]^\top$, $f_a^{-1} \triangleq [[f_x]_{l_{ds}, l_{cc}}^{-1}(x)^\top, f_{o_{st}}^{-1}(o_{st})^\top]^\top$, $b \triangleq [x^\top, o_{ct}^\top]^\top$, $f_b^{-1} \triangleq [[f_x]_{l_{sc}, l_{ds}}^{-1}(x)^\top, f_{o_{ct}}^{-1}(o_{ct})^\top]^\top$, $c \triangleq [y^\top, o_{ct}^\top]^\top$, $f_c^{-1} \triangleq [[f_y]_{s_c}^{-1}(y)^\top, f_{o_{ct}}^{-1}(o_{ct})^\top]^\top$.

(2) Identifying. This step aims to establish the validity of Equation 9, which asserts the identifiability of each latent factors. Firstly, we present the process for separating these variables. Subsequently, we will transform the resulting equations to derive Equation 9.

Considering that we have sufficient number k_u of different values of u. Taking the logarithm on the both sides of Equations 29-35, then we plug these different u (i.e., $u_0, u_1, ...u_{k_u}$) into each equation. Subtracting the first equation (containing u_0) from the second equation (u_1) to the last equation (u_{k_u}), we obtain k_u different equations for each of Equations 29-35, indexing by $q = 1, 2, \ldots, k_u$:

$$\sum_{l \in Pa(o)} \left[\langle \mathbf{T}^l(f_o^{-1}(o)), \overline{\boldsymbol{\lambda}^l}(u_q) \rangle + \sum_i \log \frac{Z_i^l(u_0)}{Z_i^l(u_q)} \right]$$

$$= \sum_{l \in Pa(o)} \left[\langle \tilde{\mathbf{T}}^l(\tilde{f}_o^{-1}(o)), \overline{\tilde{\boldsymbol{\lambda}}^l}(u_q) \rangle + \sum_i \log \frac{\tilde{Z}_i^l(u_0)}{\tilde{Z}_i^l(u_q)} \right]. \tag{36}$$

In Equation 36, o represents both observed variables and the variable pairs, i.e., $o \in \{x, y, o_{st}, o_{ct}, a, b, c\}$, where a, b, c are the variable pairs defined in the first step. When o represents variable pair, such as $a = [x^\top, o_{st}^\top]$, then $Pa(o = a) = Pa(x) \cup Pa(o_{st})$. And we define $\overline{\boldsymbol{\lambda}^l}(u_q) \triangleq \boldsymbol{\lambda}^l(u_q) - \boldsymbol{\lambda}^l(u_0)$. In order to further simplify Equation 36, we define $\mathbf{w}_q^l \triangleq \sum_i \frac{\tilde{Z}_i^l(u_0)Z_i^l(u_q)}{\tilde{Z}_i^l(u_q)Z_i^l(u_0)}$. Then we rewrite these equations in matrix form:

$$\sum_{l \in Pa(o)} \left[L^{l,\top}\mathbf{T}^l(f_o^{-1}(o)) \right] = \sum_{l \in Pa(o)} \left[\tilde{L}^{l,\top}\tilde{\mathbf{T}}^l(\tilde{f}_o^{-1}(o)) + \mathbf{w}^l \right]. \tag{37}$$

We denote Equation 37 as Eq($\cdot$). Notice that $Pa(x) = \{l_{sc}, l_{ds}, l_{cc}\}$, we will now outline the procedure for separating the latent factors in the parent set of x. By evaluating the expression $Eq(o = x) + Eq(o = o_{st}) - Eq(o = a)$, we can separate the latent factor l_{sc} of observed variable x:

$$L^{l_{sc},\top}\mathbf{T}^{l_{sc}}([f_x]_{l_{sc}}^{-1}(x)) = \tilde{L}^{l_{sc},\top}\tilde{\mathbf{T}}^{l_{sc}}([\tilde{f}_x]_{l_{sc}}^{-1}(x)) + \mathbf{w}^{l_{sc}}. \tag{38}$$

Using the same method we can separate l_{cc} of x by evaluating $Eq(o = x) + Eq(o = o_{ct}) - Eq(o = b)$:

$$L^{l_{cc},\top}\mathbf{T}^{l_{cc}}([f_x]_{l_{cc}}^{-1}(x)) = \tilde{L}^{l_{cc},\top}\tilde{\mathbf{T}}^{l_{cc}}([\tilde{f}_x]_{l_{cc}}^{-1}(x)) + \mathbf{w}^{l_{cc}}. \tag{39}$$

Afterwards, the only remaining latent factor of x, l_{ds}, is naturally separated. The above results show that for the observed variable x, all of its latent factors can be separated while preserving their individuality, without mixed information. This conclusion also holds true for other observed variables, i.e., y, o_{ct}, and o_{st}. The equations for each separated latent factors can be expressed as follows:

$$L^{l,\top}\mathbf{T}^l([f_o]_l^{-1}(o)) = \tilde{L}^{l,\top}\tilde{\mathbf{T}}^l([\tilde{f}_o]_l^{-1}(o)) + \mathbf{w}^l, \tag{40}$$

where $o \in \{x, y, o_{st}, o_{ct}, a, b, c\}$, $l \in Pa(o)$.

Based on Equation 40, we will demonstrate the validity of Equation 9. Since number k_u is enough to ensure matrix $L^{l,\top}$ has full rank, we multiply it's inverse matrix on both sides of Equation 40:

$$\mathbf{T}^l([f_o]_l^{-1}(o)) = \mathbf{A}^l\tilde{\mathbf{T}}^l([\tilde{f}_o]_l^{-1}(o)) + \mathbf{v}^l, \tag{41}$$

where $\mathbf{A}^l = (L^{l,\top})^{-1}\tilde{L}^{l,\top}$, $\mathbf{v}^l = (L^{l,\top})^{-1}\,\mathbf{w}^l$. Notice that Equation 41 is already in the same form as Equation 9.

The remaining task is to prove that the matrix $\mathbf{A}^l$ is an invertible permutation matrix, which can be achieved by directly applying Lemma 3, Theorem 2, and Theorem 3 from [25]. □

6 EXPERIMENTAL SETTINGS

6.1 Datasets

We conduct experiments on two public text summarization datasets in English: (1) **XSUM** [43] contains BBC articles accompanied with single sentence summaries (training/validation/testing size are 204,045/11,332/11,334 respectively); and (2) **CNN/DM** [19] contains news articles from CNN and Daily Mail websites paired with multi-sentence human generated summaries (training/validation/testing size are 286,817/13,368/11,487 respectively).

6.2 Evaluation Methodology

- **Automatic Evaluation:** We adopt **Rouge scores** [30] to automatically evaluate the quality of the summaries generated by our model and the baselines. Specifically, we use *Rouge-1 (R1)*, *Rouge-2 (R2)* and *Rouge-L (RL)* to measure the the uni-gram, bi-gram and longest-common subsequence similarities, respectively.
- **Human Evaluation:** We measure the **Informativeness**, **Faithfulness**, and **Fluency** referring to [13, 22, 27]. Each summary is rated on a 5-point Likert scale (higher better), to measure whether the generated summary can satisfy: (i) *Informativeness*, covering core information (i.e., the most necessary pieces) of the source document and excluding side information that may mislead the understanding of the main idea of the document; (ii) *Faithfulness*, containing only information present in the document, without introducing any made-up facts (i.e., hallucination [67]); (iii) *Fluency*, being natural and grammatically correct. Specifically, we ask three college students to score 200 samples randomly picked from the test set of CNN/DM and XSUM (100 for each).

6.3 Baselines

We compare CI-Seq2Seq against several recently proposed baseline methods: (i) **Unified VAE-PGN** [10] leverages VAE to eliminate non-critical information at a sentence-level for abstractive summarization. (ii) **VHTM** [15] jointly accomplishes summarization with topic inference via variational encoder-decoder. (iii) **T5** [54] is a pre-trained framework that converts all text-based language problems into a text-to-text format. (iv) **BART** [28] is a denoising autoencoder for pre-training Seq2Seq models. (v) **GLM** [12] is a General Language Model pre-trained with autoregressive blank infilling. (vi) **PtLAAM** [34] uses a length-aware attention mechanism to generate summaries with desired length. (vii) **PEGASUS** [74] is a pre-trained model tailored for abstractive summarization, with Gap Sentences Generation (GSG) as pre-training objective.

6.4 Implementation Details

The proposed CI-Seq2Seq can be adapted to other Seq2Seq PLMs. Here, we choose BART-large and PEGASUS-large for initialization, denoted as CI-Seq2Seq^{bart} and CI-Seq2Seq^{pega}, where the hidden size d_h is 1024, and the vocabulary size d_v is 50265 or 96103 for CI-Seq2Seq^{bart} and CI-Seq2Seq^{pega}, respectively. BART is chosen for its outstanding performance as well as less computing cost than its peers [34], and PEGASUS is chosen for its state-of-the-art performance in summarization. The number of new parameters added to CI-Seq2Seq compared to backbones is about 256M.

For hyper parameters, we use grid search to automatically find the best setup based on the validation set. We select d_u as 16 from $[8, 32]$, k_u as 5 from $[1, 20]$, and th as 0.25 from $[0.02, 0.3]$. We choose d_{ds} and d_{ss} as 128 from $\{128, 256\}$, d_{sc} as 256 from $\{256, 512\}$, and d_{cc} as 128 from $\{128, 256\}$. Note that the dimension of the SC representations d_{sc} is set larger than that of the CC representations d_{cc}, for it is very likely that the SC representations include more diverse information than the CC representations describing the core information. During training, we select λ_{kl} and λ_{lda} as 1 from $[1e^{-3}, 1]$. The batch size is searched from $\{256, 512\}$, and the learning rate is searched from $[1e^{-5}, 1e^{-4}]$. During test, we select the best number of candidate points in the range of $[5, 20]$ and the best optimization steps in the range of $[20, 100]$. $\lambda_{cc}, \lambda_{sc}, \lambda_{ds}$ are searched from $[0.001, 0.5]$, batch size is searched from $[1, 4]$, and learning rate is searched from $[0.001, 0.5]$.

Adam optimizer is utilized at both stages. We train our model on one NVIDIA Tesla V100 32GB GPUs for about 5k~10k steps for each dataset, which takes approximately six days. All experimental results are reported on the test set. Note that for baseline methods, we reproduce and evaluate our backbone models (i.e., BART and PEGASUS) ourselves to provide a fair comparison, while we report scores of other baselines from the papers. For BART, the results reproduced ourselves are almost consistent with those of the original paper [28]. For PEGASUS, the results on XSUM are almost consistent. However, our reproduced results and the reported results in the original paper [74] have a gap[2]. The result difference between this work and the original paper may come from our restriction on the maximum sequence length, which is set to 512 for the source documents and 64 for the summaries.

7 EXPERIMENTAL RESULTS

We aim to answer four research questions: **(RQ1)** Does CI-Seq2Seq enhance prediction performance on in-domain datasets? **(RQ2)** Does CI-Seq2Seq enhance generalization ability on out-of-domain datasets? **(RQ3)** Is CI-Seq2Seq Interpretable? **(RQ4)** How do latent factors and their constraints affect the performance of CI-Seq2Seq? For each question, we conduct experiments as follow.

7.1 In-domain Prediction Performance

To answer **RQ1**, we compare CI-Seq2Seq with various strong baselines on the test set of CNN/DM and XSUM, where models are trained on the training set of the same corpus.

7.1.1 *Automatic Evaluation.*

We have the following observations from Table 1: (i) VAE-based neural summarization models (i.e., *Unified VAE-PGN* and *VHTM*) perform well by automatically learning text representations containing critical information of documents. (ii) The improvements of PLMs (i.e., *T5*, *BART* and *GLM*) over previous methods demonstrate the utility of pre-training on massive corpora for downstream summarization tasks. (iii) By incorporating length-aware attention mechanism, *PtLAAM* could further enhance the performance of BART. (iv) *PEGASUS* outperforms all baselines on XSUM, showing the power of its tailored pre-training objective for summarization. On CNN/DM, PEGASUS performs less well than models initialized with BART under the same maximum sequence length constraint. The reason may lie in the different matching degree between the pre-training objective and the downstream datasets. Specifically, BART's denoising objective is to reconstruct full text, while the GSG objective for PEGASUS is to reconstruct corrupted text. Consequently, BART, with its longer target text, can better handle the long summaries of CNN/DM than PEGASUS.

When we look at our CI-Seq2Seq model, we can find that: (i) Our *CI-Seq2Seq* implemented by both BART and PEGASUS can outperform all the baselines on two datasets. For example, CI-Seq2Seq^{pega} performs 12.98% better than PEGASUS on XSUM in terms of RL. This indicates the insufficiency of only modeling statistical dependence and the effectiveness of modeling the causal relationships between observed documents and summaries. (ii) Between them, CI-Seq2Seq^{pega} performs better on XSUM, while CI-Seq2Seq^{bart}

[2] Our reproduced results were consistent with directly leveraging the checkpoint from https://huggingface.co/google/pegasus-cnn_dailymail.

Table 1: In-domain performance comparisons between our CI-Seq2Seq and the baselines on XSUM and CNN/DM datasets. Best results are marked in boldface. * indicates statistically significant improvements over baselines (p-value < 0.05).

Method	XSUM			CNN/DM		
	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
Unified VAE-PGN [10]	-	-	-	39.32	17.07	29.43
VHTM [15]	-	-	-	40.57	18.05	37.18
T5 [54]	-	-	-	42.50	20.68	39.75
BART	45.02	21.65	36.56	43.84	20.95	40.92
GLM [12]	45.50	23.50	37.30	43.80	21.00	40.50
PtLAAM [34]	45.53	21.82	36.85	44.21	20.77	40.97
PEGASUS	47.11	24.32	38.98	42.23	20.01	38.92
CI-Seq2Seq^{bart}	48.17*	25.41*	40.24*	**45.05***	22.01*	**41.96***
CI-Seq2Seq^{pega}	**51.07***	**28.68***	**44.04***	44.48	**22.88***	41.30

performs better on CNN/DM. Under the same fine-tuning setting, the possible explanation aligns with that accounting for the performance difference between BART and PEGASUS.

7.1.2 Human Evaluation. As shown in Table 2, we can observe that: (i) **Informativeness**: CI-Seq2Seq models implemented by two backbones perform better than baselines. It indicates that introducing causality helps to extract the core information into summaries, meanwhile effectively reducing the interference of side information. This is consistent with our purpose of distinguishing core content from side one in the document and leveraging the causal part for summary generation. (ii) **Faithfulness**: CI-Seq2Seq models also outperform baselines, indicating that our method could alleviate the hallucination by pursuing only core information in the document, though the hallucination problem is not our focus and deserves further exploration. (iii) **Fluency**: CI-Seq2Seq models are comparable to baselines, indicating that our models can retain the ability to generate fluent text while removing non-essential information.

Table 2: Average scores of human evaluation about Informativeness (Info.), Faithfulness (Faith.) and Fluency (Flu.). The consistency between annotators is measured by Fleiss's kappa, which is 0.71.

Method	XSUM			CNN/DM		
	Info.	Faith.	Flu.	Info.	Faith.	Flu.
BART	3.25	3.99	4.86	2.73	4.95	4.78
PEGASUS	3.57	4.21	4.92	3.13	4.95	4.88
CI-Seq2Seq^{bart}	4.01	4.35	4.91	**3.62**	4.96	**4.92**
CI-Seq2Seq^{pega}	**4.03**	**4.37**	**4.97**	3.33	**5.00**	4.90

7.2 Out-of-domain Generalization Ability

To answer **RQ2**, we compare model performance on unseen corpus under the zero-shot setting. That is, given a model trained on XSUM, we evaluate its performance on the out-of-domain (OOD) test examples from CNN/DM and vice versa. Specifically, we sample 2000 examples from each test set for evaluation.

As shown in Table 3, we observe that: though all the models struggle with OOD test examples, CI-Seq2Seq outperform baselines. For example, when training on XSUM and testing on CNN/DM, CI-Seq2Seq^{pega} beats PEGASUS by 11.55% in terms of RL. These results demonstrate that capturing the invariant causal relationships can empower the summarization model with generalization ability.

Table 3: OOD performance comparisons in terms of the generalization ability on unseen corpus. Best results are marked in boldface. * indicates statistically significant improvements over baselines (p-value < 0.05).

Train→Test	XSUM→CNN/DM			CNN/DM→XSUM		
	R1	R2	RL	R1	R2	RL
BART	25.10	6.87	17.86	21.42	3.78	14.29
PEGASUS	28.14	9.76	19.83	20.87	3.88	14.00
CI-Seq2Seq^{bart}	25.49	7.61	18.32	**23.92***	**4.86***	**15.52***
CI-Seq2Seq^{pega}	**30.37***	**11.24***	**22.12***	21.75	4.05	14.48

7.3 Interpretability of Latent Factors

To answer **RQ3**, we analyze the roles of content factors and style factors through case study and visual analysis.

Content Factors Analysis. To understand the influence of the CC and SC factors, we compare the top-3 attended words in the document when generating each token of the summary and document, based on the cross attention weights of Transformer. As shown in Table 4, summary generation guided by $\mathbf{h}_{cc}$ prefers the tokens ($\mathbf{Attended}_S$) conveying the core information of the document, e.g., "shale" and "safely", while document reconstruction guided by $\mathbf{h}_{sc}$ and $\mathbf{h}_{cc}$ attends to inessential words ($\mathbf{Attended}_D$), e.g., "involves" and "acking". Without $\mathbf{h}_{sc}$, the generated summary only captures the core content "safe" and "strengthen regulations", omitting the side content "protect public health" which exhibits frequent co-occurrence with "safe" in the corpus. This case indicates that the learned representations $\mathbf{h}_{cc}$ and $\mathbf{h}_{sc}$ can mimic the CC and SC factors to capture the core and side content in the document, respectively. We also visually analyze the learned CC and SC representations. Specifically, we randomly sample 2000 test examples from XSUM and CNN/DM respectively, and then apply t-SNE [61] to visualize $\mathbf{h}_{doc}$, $\mathbf{h}_{cc}$ and $\mathbf{h}_{sc}$. As shown in Figure 3, we can observe that: (i) The distributions of both $\mathbf{h}_{cc}$ and $\mathbf{h}_{sc}$ are smoother than that of $\mathbf{h}_{doc}$, indicating that by splitting the mixed information into distinct parts, each part will contain purer information. (ii) The distribution of $\mathbf{h}_{cc}$ exhibits higher uniformity, whereas $\mathbf{h}_{sc}$ demonstrates greater scattering. This observation indicates that the SC factors capture diverse side information for document generation and thus are dispersive.

Table 4: An example (No.8) from the XSUM test data, to analyze the roles of content factors (CC and SC) and style factors (DS and SS). We mark the core content in blue and the side content in red.

Document: ...The joint report from the Royal Society and Royal Academy of Engineering say the technique is safe if firms follow best practice and rules are enforced..."Our main conclusions are that the environmental risks of hydraulic fracturing for shale can be safely managed provided there is best practice observed and provided it's enforced through strong regulation,"...
Ground-truth summary: A gas extraction method which triggered two earth tremors near Blackpool last year should not cause earthquakes or contaminate water but rules governing it will need tightening, experts say.
BART: Shale gas extraction can be carried out safely in the UK, but stronger regulations are needed to protect public health, a report says.
CI-Seq2Seq: Shale gas extraction in the UK can be relatively safe, but the government should strengthen regulations, say scientists.
Attended$_S$: (strengthening, environmental, shale), (strengthening, environmental, strong), (technique, safely, regulations), (technique, extraction, exploration), (technique, and, environmental), (being, moot, safely), (earth, small, tremors)... **Attended$_D$:** (exploratory, fracking, involves), (involves, fracking, scientist), (fracking, acking, involves), (acking, atory, to), (involves, and, is), (involves, say, from), (and, or, into)...
CR=0.1: The environmental risks of fracking for shale gas in the UK are "very low", according to a new report. **CR=0.4:** The use of fracking to extract shale gas in the UK can be safely done, according to a new report. **CR=0.7:** Fracking, the controversial technique used to extract shale gas, can be safely done in the UK, according to a new report.

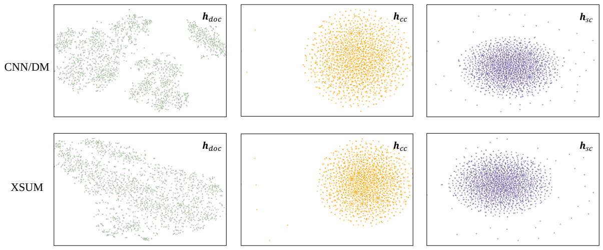

Figure 3: The t-SNE plot of $\mathbf{h}_{doc}$, $\mathbf{h}_{cc}$ and $\mathbf{h}_{sc}$ learned by ours.

Style Factors Analysis. To understand the influence of the DS and SS factors, we vary *CR* between them to explicitly control the summary generation. Specifically, we vary *CR* from 0.1 to 0.7 to change $\mathbf{h}_{ss}$. As shown in Table 4, the generated summary is concise when $CR = 0.1$, only including the necessary information, e.g., "low" and "risks". When $CR = 0.7$, the generated summary contains more specific description, e.g., "the controversial technique". The summary becomes more detailed as *CR* increases. Note that the goal of controlled generation here is not precise length control, but to control the style of the summary by utilizing *CR* as weakly-supervised signal. The results indicate that $\mathbf{h}_{ss}$ can mimic the SS factors to actively control the writing style of the summary.

7.4 Impact of Latent Factors and Constraints

To answer **RQ4**, we perform ablations on XSUM to analyze injection ways of latent factors as well as the necessity of confounder information and the content/style guidance served as constraints.

Impact of Latent Factors. We removed the addition of h_x/h_y, and the replacement of the first token in the decoder, respectively. As shown in the middle of Table 5, we found that both addition and replacement operations contribute the prediction performance, and the replacement of the starting token matters more.

Impact of Constraints. For confounder, we set $k_u = 1$ to eliminate its information. For content, we remove the content guidance loss in Equation 20. For style, we sample l_{ss} in the same way as l_{ds} without additional bridge between them. As shown in the bottom of Table 5, removing constraints on either the confounder or content/style guidance hurts the prediction performance. This demonstrates the necessity of all constraints, which is consistent with our theory that they are essential for identifying latent factors.

Table 5: Ablations of injection ways of latent factors (middle) as well as their constraints (bottom) on the subset of XSUM.

Model	R1	R2	RL
CI-Seq2Seq	41.4	17.75	33.87
w/o addition	40.97	17.78	33.77
w/o replacement	36.00	14.84	28.74
w/o confounder information	40.46	17.60	33.45
w/o content guidance	40.87	17.47	33.14
w/o style guidance	39.74	16.55	32.21

8 CONCLUSION

In this paper, we presented a principled causal perspective for text summarization. Theoretically, we proved the identifiability of the causal and non-causal factors in SCM to ensure these latent factors to be separated. Inspired by the identifiability theory, we proposed CI-Seq2Seq to learn causal representations that could mimic the causal factors for summary generation. We hope the paradigm can illuminate a promising technical direction of causality in NLP.

One limitation of our method is the slightly higher computational cost than the original Seq2Seq architecture, due to the introduction of additional parameters and the optimization procedure during inference. To address this, we plan to reduce the dimension of latent representations and explore other optimization tools. We also want to explore diverse ways to utilize confounder information and define causal factors, which can better showcase our model's strengths under the identifiability guarantees. Besides, we are interested in inducing the causal structure into extractive summarization, and exploring the controllability on more aspects.

ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 62006218, the China Scholarship Council under Grants No. 202104910234, the CAS Project for Young Scientists in Basic Research under Grant No. YSBR-034, the Innovation Project of ICT CAS under Grants No. E261090, the project under Grants No. JCKY2022130C039 and the Lenovo-CAS Joint Lab Youth Scientist Project.

REFERENCES

[1] Kartik Ahuja, Divyat Mahajan, Vasilis Syrgkanis, and Ioannis Mitliagkas. 2022. Towards efficient representation identification in supervised learning. In *CLeaR*.

[2] Naomi Altman and Martin Krzywinski. 2015. Points of Significance: Association, correlation and causation. *Nature methods* (2015).

[3] Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline Generation Based on Statistical Translation. In *ACL*.

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent Dirichlet Allocation. In *NIPS*.

[5] Léon Bottou, Jonas Peters, Joaquin Quiñonero Candela, Denis Xavier Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Y. Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: the example of computational advertising. *J. Mach. Learn. Res.* 14, 1 (2013).

[6] Boxi Cao, Hongyu Lin, Xianpei Han, Fangchao Liu, and Le Sun. 2022. Can Prompt Probe Pretrained Language Models? Understanding the Invisible Risks from a Causal View. In *ACL*.

[7] Jiaao Chen and Diyi Yang. 2021. Simple Conversational Data Augmentation for Semi-supervised Abstractive Dialogue Summarization. In *EMNLP*.

[8] Wenqing Chen, Jidong Tian, Yitian Li, Hao He, and Yaohui Jin. 2021. De-Confounded Variational Encoder-Decoder for Logical Table-to-Text Generation. In *ACL-IJCNLP*.

[9] Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He, and Yaohui Jin. 2020. Exploring Logically Dependent Multi-task Learning with Causal Inference. In *EMNLP*.

[10] Hyungtak Choi, Lohith Ravuru, Tomasz Dryjański, Sunghan Rye, Donghyun Lee, Hojung Lee, and Inchul Hwang. 2019. VAE-PGN based Abstractive Model in Multi-stage Architecture for Text Summarization. In *INLG*.

[11] John Dougrez-Lewis, Maria Liakata, Elena Kochkina, and Yulan He. 2021. Learning Disentangled Latent Topics for Twitter Rumour Veracity Classification. In *ACL-IJCNLP*.

[12] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *ACL*.

[13] Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics* 9 (2021).

[14] Angela Fan, David Grangier, and Michael Auli. 2018. Controllable Abstractive Summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*.

[15] Xiyan Fu, Jun Wang, Jinghan Zhang, Jinmao Wei, and Zhenglu Yang. 2020. Document Summarization with VHTM: Variational Hierarchical Topic-Aware Mechanism. In *AAAI-EAAI*.

[16] Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-Up Abstractive Summarization. In *EMNLP*.

[17] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer.* John Wiley & Sons.

[18] Udo Hahn and Inderjeet Mani. 2000. The Challenges of Automatic Summarization. *Computer* 33, 11 (2000).

[19] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *NIPS*.

[20] Wei-Ning Hsu, Yu Zhang, and James R. Glass. 2017. Unsupervised Learning of Disentangled and Interpretable Representations from Sequential Data. In *NeurIPS*.

[21] Zhiting Hu and Li Erran Li. 2021. A Causal Lens for Controllable Text Generation. In *NeurIPS*.

[22] Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward. In *ACL*.

[23] Dominik Janzing, Jonas Peters, Joris M. Mooij, and Bernhard Schölkopf. 2009. Identifying confounders using additive noise models. In *UAI*.

[24] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *WSDM*.

[25] Ilyes Khemakhem, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. 2020. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. In *AISTATS*, Vol. 108.

[26] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

[27] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural Text Summarization: A Critical Evaluation. In *EMNLP-IJCNLP*.

[28] 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 *ACL*.

[29] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. In *EMNLP*.

[30] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*.

[31] Chang Liu, Xinwei Sun, Jindong Wang, Haoyue Tang, Tao Li, Tao Qin, Wei Chen, and Tie-Yan Liu. 2021. Learning Causal Semantic Representation for Out-of-Distribution Prediction. In *NeurIPS*.

[32] Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward Abstractive Summarization Using Semantic Representations. In *NAACL*.

[33] Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. 2018. Generative Adversarial Network for Abstractive Text Summarization. In *AAAI-EAAI*.

[34] Yizhu Liu, Qi Jia, and Kenny Zhu. 2022. Length Control in Abstractive Summarization by Pretraining Information Selection. In *ACL*.

[35] Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. 2022. Invariant Causal Representation Learning for Out-of-Distribution Generalization. In *ICLR*.

[36] Mani Maybury. 1999. *Advances in automatic text summarization.* MIT press.

[37] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*.

[38] Jovana Mitrovic, Brian McWilliams, Jacob C. Walker, Lars Holger Buesing, and Charles Blundell. 2021. Representation Learning via Invariant Causal Mechanisms. In *ICLR*.

[39] Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. 2020. Causal Interpretability for Machine Learning - Problems, Methods and Evaluation. *SIGKDD Explor.* 22, 1 (2020).

[40] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *AAAI*.

[41] Guoshun Nan, Jiaqi Zeng, Rui Qiao, Zhijiang Guo, and Wei Lu. 2021. Uncovering Main Causalities for Long-tailed Information Extraction. In *EMNLP*.

[42] Sharmila Reddy Nangi, Niyati Chhaya, Sopan Khosla, Nikhil Kaushik, and Harshit Nyati. 2021. Counterfactuals to Control Latent Disentangled Text Representations for Style Transfer. In *ACL-IJCNLP*.

[43] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *EMNLP*.

[44] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *NAACL*.

[45] Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining text data*. Springer.

[46] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual VQA: A Cause-Effect Look at Language Bias. In *CVPR*.

[47] Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A Deep Reinforced Model for Abstractive Summarization. In *ICLR*.

[48] Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika* (1995).

[49] Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics surveys* (2009).

[50] Judea Pearl. 2009. *Causality.* Cambridge university press.

[51] Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress* (2000).

[52] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms.* The MIT Press.

[53] Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. 2021. Counterfactual Inference for Text Classification Debiasing. In *ACL-IJCNLP*.

[54] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020).

[55] Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. HiStruct+: Improving Extractive Text Summarization with Hierarchical Structure Information. In *ACL*.

[56] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* (1974).

[57] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *EMNLP*.

[58] Bernhard Schölkopf. 2022. Causality for machine learning. In *Probabilistic and Causal Inference: The Works of Judea Pearl*.

[59] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris M. Mooij. 2012. On causal and anticausal learning. In *ICML*.

[60] Xinwei Sun, Botong Wu, Xiangyu Zheng, Chang Liu, Wei Chen, Tao Qin, and Tie-Yan Liu. 2021. Recovering Latent Causal Factor for Generalization to Distributional Shifts. In *NeurIPS*.

[61] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* (2008).

[62] Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. 2021. Counterfactual Invariance to Spurious Correlations in Text Classification. In *NeurIPS*.

[63] Hong Wang, Xin Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Self-Supervised Learning for Contextualized Extractive Summarization. In *ACL*.

[64] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal Representation Learning for Out-of-Distribution Recommendation. In *WWW*.

[65] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-Agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System. In *KDD*.

[66] Congying Xia, Caiming Xiong, Philip Yu, and Richard Socher. 2020. Composed Variational Natural Language Generation for Few-shot Intents. In *EMNLP*.

[67] Yijun Xiao and William Yang Wang. 2021. On Hallucination and Predictive Uncertainty in Conditional Language Generation. In *EACL*.

[68] Yuexiang Xie, Fei Sun, Yang Deng, Yaliang Li, and Bolin Ding. 2021. Factual Consistency Evaluation for Text Summarization via Counterfactual Estimation. In *EMNLP*.

[69] Shusheng Xu, Xingxing Zhang, Yi Wu, Furu Wei, and Ming Zhou. 2020. Unsupervised Extractive Summarization by Pre-training Hierarchical Transformers. In *EMNLP*.

[70] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. 2021. A Survey on Causal Inference. *ACM Trans. Knowl. Discov. Data* 15, 5 (2021).

[71] Jen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, and I-Heng Meng. 2005. Text summarization using a trainable summarizer and latent semantic analysis. *Information processing & management* 41, 1 (2005).

[72] David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. In *HLT-NAACL*.

[73] Jichuan Zeng, Jing Li, Yulan He, Cuiyun Gao, Michael R. Lyu, and Irwin King. 2019. What You Say and How You Say it: Joint Modeling of Topics and Discourse in Microblog Conversations. *Transactions of the Association for Computational Linguistics* 7 (2019).

[74] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *ICML*, Vol. 119.

[75] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. 2021. Deep Stable Learning for Out-of-Distribution Generalization. In *CVPR*.

[76] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR*.

[77] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *WWW*.

[78] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive Summarization as Text Matching. In *ACL*.

[79] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for Effective Neural Extractive Summarization: What Works and What's Next. In *ACL*.

[80] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural Document Summarization by Jointly Learning to Score and Select Sentences. In *ACL*.

[81] Yicheng Zou, Hongwei Liu, Tao Gui, Junzhe Wang, Qi Zhang, Meng Tang, Haixiang Li, and Daniell Wang. 2022. Divide and Conquer: Text Semantic Matching with Disentangled Keywords and Intents. In *ACL*.

[82] Yanyan Zou, Xingxing Zhang, Wei Lu, Furu Wei, and Ming Zhou. 2020. Pretraining for Abstractive Document Summarization by Reinstating Source Text. In *EMNLP*.