Compositional Generalization for Multi-Label Text Classification: A Data-Augmentation Approach

Yuyang Chai[1,*], Zhuang Li[2,*], Jiahui Liu[1], Lei Chen[1], Fei Li[1], Donghong Ji[1], Chong Teng[1,†]

[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University
[2]Faculty of Information Technology, Monash University
{yychai, liujiahui, chhenl, lifei_csnlp, dhji, tengchong}@whu.edu.cn, zhuang.li1@monash.edu

Abstract

Despite significant advancements in multi-label text classification, the ability of existing models to generalize to novel and seldom-encountered complex concepts, which are compositions of elementary ones, remains underexplored. This research addresses this gap. By creating unique data splits across three benchmarks, we assess the compositional generalization ability of existing multi-label text classification models. Our results show that these models often fail to generalize to compositional concepts encountered infrequently during training, leading to inferior performance on tests with these new combinations. To address this, we introduce a data augmentation method that leverages two innovative text generation models designed to enhance the classification models' capacity for compositional generalization. Our experiments show that this data augmentation approach significantly improves the compositional generalization capabilities of classification models on our benchmarks, with both generation models surpassing other text generation baselines. Our codes available at https://github.com/yychai74/LD-VAE.

Introduction

Multi-label text classification (MLTC) involves identifying the labels associated with the input text. This task has broad applications in natural language processing (NLP), including sentiment analysis in tweets (Mohammad et al. 2018), subject identification in interdisciplinary academic articles (Yang et al. 2018), and movie genre classification based on movie reviews (Maiya 2019). Although there has been significant progress in improving classifier performance across various MLTC benchmarks, whether existing MLTC models can generalize compositionally has received limited attention in prior MLTC studies.

Compositional generalization (CG) is a fundamental ability inherent to human intelligence, enabling the recognition of novel and infrequently occurring high-level concepts that are compositions of more atomic elements (Chomsky 2014). For example, once a person learns to understand the emotions of joy and sadness in simple phrases like 'I am sad' and 'I rejoice', respectively, he can effortlessly recognize a

---

*These authors contributed equally.
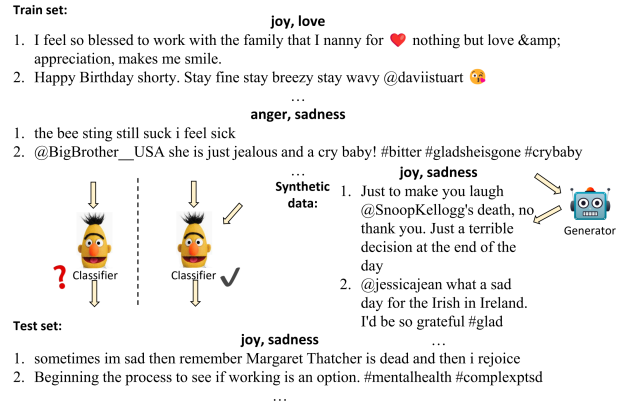
†Corresponding author.

Figure 1: Illustration of the CG challenge in MLTC and an overview of our proposed data augmentation solution.

complex emotion in a tweet such as 'sometimes I am sad, then remember Margaret is dead, and then I rejoice'. This tweet conveys a nuanced composite of two emotions occurring simultaneously. In contrast to humans, our initial research indicates that current MLTC models struggle to identify these nuanced compositions if they infrequently occur in the training set. The T5-based (Raffel et al. 2020) MLTC model (Chai et al. 2022), for example, only achieved less than 2% accuracy on the SemEval test set (Mohammad et al. 2018) for previously unseen emotional compositions, despite ample training data for each elementary emotion. Conversely, when the training data contains abundant emotional compositions as those found in the test set, its accuracy exceeded 28%. This discrepancy underscores the urgent need for MLTC models capable of generalizing to novel compositions, making them more effective in a world that continuously presents new composite knowledge.

This study offers the first in-depth exploration of the CG challenges that impact MLTC. We utilize three MLTC benchmarks that tackle three tasks: emotion classification, subject identification of abstracts, and genre classification of movie reviews. Traditional MLTC benchmarks typically employ random splits, where all the compositions of individual labels are prevalent across both training and test sets. This methodology hinders a rigorous evaluation of the models' capacity to generalize compositionally. Inspired by the

inherent human ability to recognize unfamiliar composite concepts with minimal exposure, as well as informed by previous research on CG (Keysers et al. 2019; Finegan-Dollak et al. 2018), we propose a distinct data split to evaluate the CG capabilities of MLTC models. All elementary labels are abundant in the training set in this split, but instances with novel label compositions in the test set are seldom found in the training data. To enhance this evaluation, we introduce two novel metrics that evaluate model performance in terms of compositional rather than individual label predictions.

Compositional data augmentation is a widely-adopted strategy for enhancing the CG ability of machine learning models in fields such as semantic parsing (Qiu et al. 2022a; Yang, Zhang, and Yang 2022; Andreas 2020) and few-shot single-label classification (Li et al. 2022). This strategy augments training sets with instances of seldom-occurring compositions. Inspired by the methodology, we introduce an innovative data augmentation technique specifically designed for MLTC. Our approach consists of three key components: i) a model that learns the distribution of target label compositions using limited examples, ii) a conditional generation model that synthesizes new text instances conditioned on novel label compositions drawn from the learned distribution, and iii) a filter to discard invalid examples.

A fundamental challenge is devising a generation model that can identify and systematically combine phrases tied to individual labels, thereby forming coherent text that reflects the given label composition. A significant hurdle is the entangled representations of individual semantic factors in neural sequence models, hindering the alignment learning of labels with corresponding text fragments. Therefore, We propose two innovative representation disentanglement solutions for conditional generation models. The first, Label-specific Prefix-Tuning (LS-PT), uses label-specific prefix vectors (Li and Liang 2021) as disentangled label representations. The second, Label Disentangled Variational Autoencoder (LD-VAE), employs disentanglement learning and variational autoencoders (Kingma and Welling 2013) to extract disentangled label representations. Conditioned on the disentangled label representations, the conditional generation models can yield high-quality instances.

Overall, our contributions are three-fold:

- We are the first to explore the critical issue of CG on three MLTC benchmarks. By introducing a unique evaluation data split and two novel evaluation metrics, we can measure the CG abilities of existing models. Our analysis reveals existing MLTC models lack CG capability.

- We propose a novel data augmentation strategy to augment instances with rare label compositions. Our empirical studies demonstrate that this approach with various generation models dramatically boosts the CG capability of MLTC models across all evaluated metrics.

- We design two generation models central to our data augmentation approach. These models focus on disentangling and composing individual label representations to generate instances associated with novel label compositions. Experiments show that both models surpass other generation baselines regarding CG evaluation metrics.

Compositional Multi-label Text Classification

Problem Setting   In MLTC, the aim is to determine a function $\pi_\theta : \mathcal{X} \to \mathcal{Y}$, that maps a text sequence $\boldsymbol{x} \in \mathcal{X}$ to its corresponding label set $\boldsymbol{y} \in \mathcal{Y}$. Here, $\boldsymbol{x} = \{x_0, ..., x_n\}$ represents a sequence of n tokens, while $\boldsymbol{y} = \{y_0, ..., y_m\}$ denotes a composition of m unordered labels in a label set.

We assume that training set samples $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathcal{D}_{train}$ are drawn from a source distribution $P_s(\boldsymbol{x}, \boldsymbol{y})$, while test set samples $\mathcal{D}_{test}$ are sourced from a target distribution $P_t(\boldsymbol{x}, \boldsymbol{y})$. In the conventional training and evaluation setting, where the dataset is split randomly, both distributions align, i.e., $P_s(\boldsymbol{x}, \boldsymbol{y}) = P_t(\boldsymbol{x}, \boldsymbol{y})$. However, in our compositional data split, which is in line with the CG studies for other tasks (Qiu et al. 2022a; Yang, Zhang, and Yang 2022; Andreas 2020), these distributions diverge. We assume the conditional distribution remains the same, i.e., $P_s(\boldsymbol{x}|\boldsymbol{y}) = P_t(\boldsymbol{x}|\boldsymbol{y})$, while the label composition distribution varies: $P_s(\boldsymbol{y}) \neq P_t(\boldsymbol{y})$. In addition, in the CG setup, the atomic individual labels $y \in Y$ are *shared and abundant* across both training and test datasets. Additionally, an optional support set $\mathcal{D}_s$ complements the training set, comprised of limited examples drawn from $P_t(\boldsymbol{x}, \boldsymbol{y})$. This aids in the few-shot learning of novel compositions, as seen in the setups of the CG works (Lee et al. 2019; Li et al. 2021b).

Concretely, the training, support, and test sets are constructed as follows. Let the complete MLTC dataset be denoted as $\mathcal{D}_{ori} = \mathcal{D}_{train} \cup \mathcal{D}'_{test}$. We partition $\mathcal{D}_{ori}$ into training and preliminary test sets based on label compositions. Specifically, $\mathcal{D}_{train} = \mathcal{X}_{train} \times \mathcal{Y}_{train}$ and $\mathcal{D}'_{test} = \mathcal{X}'_{test} \times \mathcal{Y}'_{test}$. We ensure that the training and preliminary test sets do not share any label compositions: $\mathcal{Y}_{train} \cap \mathcal{Y}'_{test} = \emptyset$. The preliminary test set contains M unique label compositions, where $|\mathcal{Y}'_{test}| = M$. We then randomly sample a subset of $\mathcal{D}'_{test}$ containing N_s examples to form the support set, denoted as $\mathcal{D}_s$. The remaining examples in $\mathcal{D}'_{test}$ constitute the actual test set, $\mathcal{D}_{test} = \mathcal{D}'_{test} \setminus \mathcal{D}_s$. Given that we employ a data augmentation module, the models within this module are trained on the union of the training and support sets, $\mathcal{D}_{cg} = \mathcal{D}_s \cup \mathcal{D}_{train}$. This data augmentation process augments $\mathcal{D}_s$ to yield a synthetic data set, denoted as $\mathcal{D}_{aug}$. Our multi-label classifier is trained on the combined training, support, and augmented sets, $\mathcal{D}_{mltc} = \mathcal{D}_s \cup \mathcal{D}_{train} \cup \mathcal{D}_{aug}$, and it is evaluated on $\mathcal{D}_{test}$.

Evaluation Metrics   To assess MLTC performance, one of the prevailing metrics is the **Jacc**ard score, as highlighted in Yang et al. (2018); Chai et al. (2022). The Jaccard score computes the ratio of correctly identified labels to the union of the two label sets for each instance. However, this metric might inadequately capture the CG capability since it measures performance based on individual labels. Hence, even if a predicted label composition is erroneous, a high Jaccard score can still be attained. To capture composition-level performance, we use **Acc**uracy metric as in Yarullin and Serdyukov (2021). This metric verifies if the predicted label set exactly matches the ground-truth set. While the Accuracy provides valuable insights at the composition level, we anticipate a more nuanced analysis. Therefore, we introduce two supplementary metrics, **Corr**ectness

and **Comp**leteness, which evaluate performance at the compositional level but with a degree of flexibility. The Correctness metric evaluates if every predicted label exists in the ground truth, whereas Completeness checks if every ground-truth label has been forecasted by the model:

$$\text{Corr}(\boldsymbol{y}_p, \boldsymbol{y}_g) = \mathbb{1}((\boldsymbol{y}_p \cap \boldsymbol{y}_g) = \boldsymbol{y}_p) \tag{1}$$

$$\text{Comp}(\boldsymbol{y}_p, \boldsymbol{y}_g) = \mathbb{1}((\boldsymbol{y}_p \cap \boldsymbol{y}_g) = \boldsymbol{y}_g) \tag{2}$$

where $\boldsymbol{y}_p$ is the predicted and $\boldsymbol{y}_g$ is the ground-truth label sets of a given instance, respectively. Overall, the aggregated performance scores for the entire test set are computed as the average of the Jaccard, Accuracy, Correctness, and Completeness scores across all test instances.

Discussion Table 1 reveals significant declines in the accuracy of existing MLTC models when transitioning from the iid to the compositional data split for training and evaluation. This performance drop highlights the limited CG capability of MLTC models. Our observations suggest that these models often face issues such as learning spurious correlations or being vulnerable to input perturbations. As seen in emotion classification, the models often predict emotions of the same polarity but struggle to accurately identify compositions that combine emotions of differing sentiment polarities, primarily because such combinations are rare in the training set. We conjecture they have learned a spurious correlation between the frequent co-occurrence of like-polarity emotions. Another challenge for the MLTC models is their inability to accurately predict the same labels corresponding to input texts with notable linguistic variations across the training and test sets. To address these challenges, data augmentation has been shown to mitigate the impact of spurious correlations (Wang and Culotta 2021) and enhance the robustness of machine learning models (Goodfellow, Shlens, and Szegedy 2014; Huang et al. 2021b).

Compositional Data Augmentation

Compositional data augmentation focuses on sampling examples from the target distribution $P_t(\boldsymbol{x}, \boldsymbol{y})$. This distribution can be factorized as $P_t(\boldsymbol{x}, \boldsymbol{y}) = P_t(\boldsymbol{y})P_t(\boldsymbol{x}|\boldsymbol{y})$. The challenge of this approach then becomes to craft two distinct models: one that models the distribution of label compositions $P_t(\boldsymbol{y})$, and another that models the conditional distribution $P_t(\boldsymbol{x}|\boldsymbol{y})$. We also introduce a quality control mechanism to filter out low-quality synthetic examples. We hypothesize that exposing MLTC models to diverse syntactic structures and phrases, associated with novel label compositions, can enhance their ability to learn true causal correlations between text and labels. Furthermore, introducing diverse linguistic variations related to each label can improve the model's robustness to input perturbations.

Label Generative Model To model the label distribution $P_t(\boldsymbol{y})$, we approach it as a sequence modelling task. The learning objective is to optimize the parameters $\boldsymbol{\theta}$ to maximize the likelihood of the sequence of tokens in concatenated label phrases within the support set:

$$\arg\max_{\boldsymbol{\theta}} \prod_{\boldsymbol{y}' \in \mathcal{Y}_s} \prod_{t=0}^{|\boldsymbol{y}'|} P_{\boldsymbol{\theta}}(y_t'|\boldsymbol{y}_{<t}') \tag{3}$$

Model	SemEval		AAPD		IMDB	
	iid	CG	iid	CG	iid	CG
BERT	27.31	2.85	37.79	22.02	26.17	4.35
BERT+P	26.88	2.71	34.22	19.08	18.93	1.95
BERT+MAGNET	24.95	2.23	37.70	14.48	21.73	3.68
BERT+SGM	19.87	2.15	37.71	15.23	18.65	3.22
BERT+DBloss	26.76	3.72	36.50	14.57	40.81	2.54
T5+CLP	28.34	1.26	42.20	9.17	39.78	1.35

Table 1: The classification accuracies of existing MLTC models on both iid and CG splits across three benchmarks.

where $\mathcal{Y}_s$ is the set of all label compositions within the support set $\mathcal{D}_s$, $\boldsymbol{y}'$ denotes the sequence of tokens in the concatenated label phrases originating from a label set $\boldsymbol{y}$ for each instance, and y_t' is a token at position t in $\boldsymbol{y}'$. To accomplish this, we fine-tune a pre-trained language model GPT2 (Radford et al. 2019) to estimate the token probability $P_{\boldsymbol{\theta}}(y_t'|\boldsymbol{y}_{<t}')$, benefiting from its pre-existing knowledge about distributions of label phrase tokens. In the practical implementation, we prepend a prefix prompt, like 'A tweet can express the following emotions:', to the label phrases during training and inference. In the zero-shot setting, where a support set is absent, we instruct GPT2 using the prompt and constrain it only to generate label-related phrases.

Conditional Text Generative Models

As we assume that in the CG split, the source and target conditional distributions align, i.e., $P_s(\boldsymbol{x}|\boldsymbol{y}) = P_t(\boldsymbol{x}|\boldsymbol{y})$, the conditional generation model, therefore, can be trained on the combination of the training and support sets $\mathcal{D}_{cg}$. The learning objective for conditional text generation becomes:

$$\arg\max_{\boldsymbol{\theta}} \prod_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}_{cg}} \prod_{t=0}^{|\boldsymbol{x}|} P_{\boldsymbol{\theta}}(x_t|\boldsymbol{x}_{<t}, \boldsymbol{y}) \tag{4}$$

A common method, as in Li et al. (2022), to implement this generation model is to fine-tune a pre-trained language model, such as GPT2 or T5. Pre-trained models appear to excel in compositional sequence generalization (Qiu et al. 2022b). During inference, the model converts the concatenated label phrases in the label set $\boldsymbol{y}$ into contextualized representations, prompting the generation model to produce text $\boldsymbol{x}$ conditioned on the representations. However, after fine-tuned on the compositionally-biased dataset $\mathcal{D}_{cg}$, the representations of different labels, as encoded by the Transformer (Vaswani et al. 2017), can be severely entangled. Each label's representation influences others, making it non-invariant to changes in its co-occurring labels. Such entanglement can hinder the generation model from identifying the true associations between each label representation and its corresponding phrases or syntactic structures in the text $\boldsymbol{x}$, compromising its ability to composite phrases or structures into high-quality texts for data augmentation. Following this, we present two models focusing on disentangling the label representations for effective text generation.

Label Specific Prefix-Tuning (LS-PT) We suspect that the entanglement observed in label presentations arises from

the innate cross-attention mechanism of Transformers. Each label presentation serves as an attended representation of all labels. To address this challenge, we introduce a novel method that encodes each label in the label composition set $y_i \in \boldsymbol{y}$ with its distinct representation, $\boldsymbol{z}_{y_i} \in \mathbb{R}^{L \times H}$. This representation is designed to be minimally influenced by neighboring labels. The concatenated representations of individual labels are processed using a multilayer perceptron (MLP) to produce a composite label representation:

$$\mathbf{M} = \text{MLP} \left(\begin{bmatrix} \boldsymbol{z}_{y_0} \\ \vdots \\ \boldsymbol{z}_{y_m} \end{bmatrix} \right). \tag{5}$$

Building upon the prefix-tuning technique (Li and Liang 2021), we estimate the conditional probability $P_{\boldsymbol{\theta}}(x_t | \boldsymbol{x}_{<t}, \boldsymbol{y})$, which is further elaborated as $P_{\boldsymbol{\theta}}(x_t | \boldsymbol{x}_{<t}, \boldsymbol{z}_{y_0}, \ldots, \boldsymbol{z}_{y_m}) = \text{softmax}(\boldsymbol{h}_i \mathbf{W})$. The hidden state $\boldsymbol{h}_i$ is defined as:

$$\boldsymbol{h}_i = \begin{cases} \mathbf{M}[i, :], & \text{if } i \in M_{idx}, \\ \text{LM}(\mathbf{M}_{\theta}, \boldsymbol{h}_{<i}), & \text{otherwise.} \end{cases} \tag{6}$$

where the composite label representation $\mathbf{M} \in \mathbb{R}^{|M_{idx}| \times H'}$ is considered the prefix matrix, $\boldsymbol{h}_i \in \mathbb{R}^{1 \times H'}$ is the hidden state of a language model, M_{idx} denotes the column indices of the prefix matrix with $|M_{idx}| = |\boldsymbol{y}| \cdot L$, and $\mathbf{LM}$ refers to a pre-trained auto-regressive language model. Here we adopt GPT2 with frozen parameters as $\mathbf{LM}$ to maximize benefits from its pre-trained compositional knowledge.

Inference. At the inference stage, we draw concatenated label phrases from $P(\boldsymbol{y}')$ and convert each set of phrases $\boldsymbol{y}'$ into a set of label ids, $\boldsymbol{y}$. Texts with new label compositions are then generated conditioned on $\boldsymbol{y}$ and label phrases $\boldsymbol{y}'$.

Label Disentangled Variational Autoencoder (LD-VAE)
The model VAE-DPrior (Li et al. 2022) aims to disentangle representations associated with labels and content, and subsequently composite these into new examples. To realize this, the model adjusts its estimation from $P_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{y})$ to $P_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{y}, c)$, where c denotes a variable capturing the prior knowledge of content related to the text $\boldsymbol{x}$. The learning objective of VAE-DPrior is to maximize the Evidence Lower Bound (ELBO) of $P_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{y}, c)$. Specifically, after introducing a latent variable $\boldsymbol{z}_c$ associated with the content c and a latent variable $\boldsymbol{z}_{\boldsymbol{y}}$ associated the label $\boldsymbol{y}$, and providing a strong conditional independence assumption, $P(\boldsymbol{z}_c, \boldsymbol{z}_{\boldsymbol{y}} | \boldsymbol{z}, c, \boldsymbol{y}) = P(\boldsymbol{z}_c | \boldsymbol{x}, c) P(\boldsymbol{z}_{\boldsymbol{y}} | \boldsymbol{x}, \boldsymbol{y})$, ELBO objective for VAE-DPrior is:

$$\begin{aligned} & \mathbb{E}_{Q_{\boldsymbol{\phi}}(\boldsymbol{z}_c, \boldsymbol{z}_{\boldsymbol{y}} | \boldsymbol{x}, c, \boldsymbol{y})}[\log P_{\boldsymbol{\theta}}(\boldsymbol{x} | \boldsymbol{z}_c, \boldsymbol{z}_{\boldsymbol{y}})] & \}\mathcal{L}_r \\ & - \mathbb{D}_{\text{KL}}(Q_{\boldsymbol{\phi}}(\boldsymbol{z}_c | \boldsymbol{x}, c) \| P_{\boldsymbol{\theta}}(\boldsymbol{z}_c | c)) & \}\mathcal{L}_c \\ & - \mathbb{D}_{\text{KL}}(Q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{y}} | \boldsymbol{x}, \boldsymbol{y}) \| P_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{y}} | \boldsymbol{y})) & \}\mathcal{L}_l \end{aligned} \tag{7}$$

where $\mathcal{L}_r$ represents the text reconstruction loss for the VAE decoder, while $\mathcal{L}_c$ and $\mathcal{L}_l$ are the regularization loss terms for content and label encoders, respectively. With sufficiently divergent prior conditional distributions $P_{\boldsymbol{\theta}}(\boldsymbol{z}_c | c)$ and $P_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{y}} | \boldsymbol{y})$ during regularization, VAE-DPrior disentangles label and content representations. However, the model overlooks disentanglement within label representations. To address this, we consider a set of representations $\boldsymbol{z}_{\boldsymbol{y}} = \{\boldsymbol{z}_{y_0}, \ldots, \boldsymbol{z}_{y_m}\}$, where each label y from the

label set $\boldsymbol{y}$ is *only* associated with a specific latent variable $\boldsymbol{z}_y$. Adopting a conditional independence assumption, given by $P(\boldsymbol{z}_{\boldsymbol{y}} | \boldsymbol{x}, \boldsymbol{y}) = \prod_{i=0}^{m} P(\boldsymbol{z}_{y_i} | \boldsymbol{x}, y_i)$ and $P(\boldsymbol{z}_{\boldsymbol{y}} | \boldsymbol{y}) = \prod_{i=0}^{m} P(\boldsymbol{z}_{y_i} | y_i)$, we update $\mathcal{L}_l$ using the chain rule of the Kullback–Leibler (KL) divergence:

$$\mathcal{L}_l = -\sum_{i=0}^{m} \mathbb{D}_{\text{KL}}(Q_{\boldsymbol{\phi}}(\boldsymbol{z}_{y_i} | \boldsymbol{x}, y_i) \| P_{\boldsymbol{\theta}}(\boldsymbol{z}_{y_i} | y_i)) \tag{8}$$

We aim to disentangle the label representations by employing distinct conditional priors for different label encoders. Since our focus is not on the deep theoretical foundations of disentanglement learning, those interested in the theory can refer to the original VAE-DPrior work. The proof of how we leverage the KL chain rule for $\mathcal{L}_l$ can be found in Appendix. Next, we delve into the implementation of LD-VAE.

Regularization for Content Encoders. In line with VAE-DPrior, the content knowledge $c \in \mathcal{C}$ derived from an input text $\boldsymbol{x}$ is represented by one of the $|\mathcal{C}|$ centroids. These centroids are formed using k-means clustering, with BERT (Devlin et al. 2019) encoding each text $\boldsymbol{x} \in \mathcal{X}_{cg}$ in $\mathcal{D}_{cg}$. Formally, the content prior $P_{\boldsymbol{\theta}}(\boldsymbol{z}_c | c)$ assumes the form of a conditional Gaussian, $\mathcal{N}(\boldsymbol{z}_c; \boldsymbol{\mu}_c^p, \lambda_c \mathbf{I})$. Here, the mean $\boldsymbol{\mu}_c^p = \boldsymbol{k}_c \mathbf{W}_c$ is a linear projection of the relevant cluster centroid vector $\boldsymbol{k}_c \in \mathbb{R}^{1 \times H}$. The text $\boldsymbol{x}$ belongs to the cluster with centroid c. I is an identity matrix determining variance.

The posterior distribution $Q_{\boldsymbol{\phi}}(\boldsymbol{z}_c | \boldsymbol{x}, c)$ is modeled by the **content encoder**, using the VAE reparameterization trick:

$$\boldsymbol{v}_c = \text{Mean}(\text{GRU}_c(\mathbf{V}_{\boldsymbol{x}})) \tag{9}$$

$$\log \boldsymbol{\sigma}_c^q, \boldsymbol{\mu}_c^q = \text{MLP}_{\boldsymbol{\sigma}_c^q}(\boldsymbol{v}_c), \text{MLP}_{\boldsymbol{\mu}_c^q}(\boldsymbol{v}_c) \tag{10}$$

$$\boldsymbol{z}_c = \boldsymbol{\mu}_c^q + \exp(\frac{1}{2} \log \boldsymbol{\sigma}_c^q) \odot \boldsymbol{\epsilon}_c \tag{11}$$

where $\odot$ is the element-wise product, $\boldsymbol{\epsilon}_c$ is Gaussian noise from the distribution $\mathcal{N}(0, \lambda_c \mathbf{I})$. $\mathbf{V}_{\boldsymbol{x}}$ is the contextualized representation of $\boldsymbol{x}$ encoded by BERT with parameters frozen, the GRU is Gated Recurrent Units (Cho et al. 2014), and $\boldsymbol{z}_c \in \mathbb{R}^{1 \times H}$ is a one-dimensional vector. Ideally, each variable c would require separate parameters, leading to $|\mathcal{C}|$ content encoders and prior conditional Gaussians with distinct sets of parameters. Given the large size of $\mathcal{C}$, we utilize shared parameters from GRUs, and MLPs across all $c \in \mathcal{C}$ for parameter efficiency. Hence, the content encoder's regularization loss $-\mathbb{D}_{\text{KL}}(Q_{\boldsymbol{\phi}}(\boldsymbol{z}_c | \boldsymbol{x}, c) \| P_{\boldsymbol{\theta}}(\boldsymbol{z}_c | c))$ is formulated as maximizing:

$$\mathcal{L}_c = -\frac{1}{2\lambda_c} \left(\|\boldsymbol{\mu}_c^q - \boldsymbol{\mu}_c^p\|^2 + (\boldsymbol{\sigma}_c^q - \lambda_c \log(\boldsymbol{\sigma}_c^q)) \cdot \mathbf{1} \right) \tag{12}$$

where $\mathbf{1} \in \mathbb{R}^{H \times 1}$ is used to sum over elements of the vector.

Regularization for Label Encoders. For each label $y_i \in \boldsymbol{y}$, the conditional prior $P_{\boldsymbol{\theta}}(\boldsymbol{z}_{y_i} | y_i)$ takes the form of a conditional Gaussian form, represented as $\mathcal{N}(\boldsymbol{z}_{y_i}; \boldsymbol{\mu}_{y_i}^p, \lambda_{y_i} \mathbf{I})$. The mean of this Gaussian corresponds to a linear projection of the embedding of the label phrase, denoted as $\boldsymbol{\mu}_{y_i}^p = \boldsymbol{l}_{y_i} \mathbf{W}_{y_i}$, with $\boldsymbol{l}_{y_i}$ being encoded using frozen BERT.

The posterior distribution $Q_{\boldsymbol{\phi}}(\boldsymbol{z}_{y_i} | \boldsymbol{x}, y_i)$ is modeled by **label encoder**, which mirrors the structure of the content encoder. Unlike the content encoder, the GRU parameters are not shared. Therefore, we apply m label encoders with distinct sets of parameters, with each set modeling the posterior distribution for one latent variable $\boldsymbol{z}_{y_i}$.

Model	SemEval				AAPD				IMDB			
	Jacc	Acc	Corr	Comp	Jacc	Acc	Corr	Comp	Jacc	Acc	Corr	Comp
No Aug.	44.90	2.92	49.23	15.31	52.24	21.96	42.38	42.54	42.94	4.48	61.09	8.77
Concat	45.84	3.25	48.37	15.14	-	-	-	-	46.13	8.71	61.02	14.25
Flan-T5	47.84	8.01	52.53	**17.32**	55.98	26.46	44.42	45.68	47.39	11.69	60.70	16.98
VAE-DPrior	47.50	6.07	49.62	17.17	55.52	24.76	42.28	46.30	46.49	9.68	59.29	15.54
GPT3.5	46.56	4.74	47.58	16.69	56.52	26.63	44.77	46.42	46.69	10.04	62.85	15.13
GPT2-PT	47.44	6.81	50.53	17.22	56.38	28.03	45.95	46.22	47.12	11.38	61.48	16.35
LS-PT	**48.02**	8.29	52.94	16.93	57.95	30.21	47.74	47.23	**48.36**	**12.56**	62.37	**17.57**
LD-VAE	47.94	**8.44**	**53.10**	16.96	**58.50**	**31.11**	**48.67**	**48.09**	48.07	11.75	**63.23**	16.92

Table 2: Classification results using BERT with augmentation instances from various generators on the CG data split.

The regularization loss for the label encoders becomes:

$$\mathcal{L}_l = \sum_{i=1}^{m} -\frac{1}{2\lambda_{y_i}} \left(\|\boldsymbol{\mu}_{y_i}^q - \boldsymbol{\mu}_{y_i}^p\|^2 + (\boldsymbol{\sigma}_{y_i}^q - \lambda_{y_i} \log(\boldsymbol{\sigma}_{y_i}^q)) \cdot \mathbf{1} \right)$$

(13)

Text Reconstruction. The reconstruction loss is the maximum likelihood loss used to optimize the parameters of the decoder. The decoder shares the same structure as LS-PT and employs prefix-tuning, allowing a GPT2 to generate text conditioned on z_c and $\{z_{y_0}, ..., z_{y_m}\}$. There is a slight difference from Eq. 5 as shown below:

$$\mathbf{M} = \text{MLP} \left(\begin{bmatrix} z'_{y_0}; z'_c \\ \vdots \\ z'_{y_m}; z'_c \end{bmatrix} \right)$$

(14)

where $z'_{y_i} \in \mathbb{R}^{L \times H}, z'_c \in \mathbb{R}^{L \times H}$ is obtained by repeating z_{y_i} and z_c L times, respectively, because we expect a longer prefix length can enhance the performance of prefix-tuning. The z'_{y_i} and z'_c are vertically concatenated together.

Inference. During inference, the encoders are discarded. The content variable c is randomly sampled from $\mathcal{C}$, based on the assumption that $P(c)$ follows a uniform distribution. The label set $\boldsymbol{y} = \{y_0, ..., y_m\}$ is drawn from the label generative model $P(\boldsymbol{y}')$. With the sampled c and $\boldsymbol{y}$, we sample latent representations z_c and z_{y_i} from the conditional Gaussian priors $P_{\boldsymbol{\theta}}(z_c|c)$ and $P_{\boldsymbol{\theta}}(z_{y_i}|y_i)$, respectively. The decoder then generate synthetic text $\boldsymbol{x}'$ conditioned on the latent label representations $\boldsymbol{y}$ and label phrases $\boldsymbol{y}'$.

Quality Control (QC) QC is implemented using a BERT-based MLTC classifier trained on $\mathcal{D}_{cg}$. We first overgenerate synthetic examples, each text $\boldsymbol{x}$ paired with a label set $\boldsymbol{y}_s$. Next, we use the classifier to predict the labels $\boldsymbol{y}_p$ for each text $\boldsymbol{x}$. We then rank the examples by their Jaccard scores, $\text{Jacc}(\boldsymbol{y}_p, \boldsymbol{y}_s)$, and retain those with the top K highest scores.

Experiments

Datasets We conduct experiments on the compositional splits of three datasets: SemEval (Mohammad et al. 2018), AAPD (Yang et al. 2018), and IMDB (Maiya 2019). During the data splitting process, we allocate 20 label compositions to the test set. After splitting, SemEval, a multi-label emotion classification dataset, comprises 9,530 training, 50 support, and 1,403 test examples. AAPD features academic paper abstracts annotated with subject categories from Arxiv

and contains 50,481 training, 50 support, and 5,309 testing examples. IMDB provides movie reviews annotated with movie genres and includes a total of 107,944 training, 50 support, and 9,200 test samples. We first overgenerate 2,000, 10,000, and 24,000 examples, and then apply quality control to filter the synthetic data down to sizes of 1,000, 5,000, and 12,000 for SemEval, AAPD, and IMDB, respectively.

MLTC Models We evaluate the performance of MLTC models on our CG tasks: *i)* **BERT (Devlin et al. 2019)** employs BERT combined with MLPs on BERT's top layers for multi-label classification and is optimized using cross-entropy loss. *ii)* **BERT+P** has the same structure as **BERT** but is optimized using p-tuning (Liu et al. 2022). *iii)* **BERT+DBloss (Huang et al. 2021d)** uses BERT and is optimized with a loss function tailored specifically to address label distribution imbalances in MLTC datasets. *iv)* **BERT+MAGNET (Pal, Selvakumar, and Sankarasubbu 2020)** integrates BERT with a graph attention network designed to learn correlations between labels. *v)* **BERT+SGM (Yang et al. 2018)** treats MLTC as a sequence generation task. The word embeddings of the text, encoded by BERT, are then fed into an LSTM that learns label correlations and generates label predictions. *vi)* **T5+CLP (Chai et al. 2022)** is a model based on T5 (Raffel et al. 2020) designed to capture label correlations using the decoder combined with a contrastive learning loss.

Generator Baselines We compare five conditional text generators trained on $\mathcal{D}_{cg}$. Each generator generates text conditioned on the same set of novel label compositions, which are sampled from the label generative model. Furthermore, all employ the same filter model for quality control. *i)* **Concat.** This method simply concatenates single-labeled instances to create synthetic examples with specific label compositions. This concept aligns with the approach taken by Jia and Liang (2016) for the semantic parsing task. Note: AAPD is not suited for this baseline because it lacks single-labeled instances. *ii)* **Flan-T5 (Chung et al. 2022)** is a sequence-to-sequence language model (Sutskever, Vinyals, and Le 2014) pre-trained on thousands of NLP tasks. It crafts text based on composites of label phrases processed by its encoder. *iii)* **VAE-DPrior (Li et al. 2022)** employs the VAE and disentanglement learning to disentangle label and content representations, and then generates new texts conditioned on a

Model	BERT		BERT+P		BERT+DBloss		BERT+MAGNET		BERT+SGM		T5+CLP	
	-	+	-	+	-	+	-	+	-	+	-	+
Jacc	52.24	58.50	49.16	53.65	50.92	58.52	42.26	51.27	43.95	53.37	43.03	43.72
Acc	21.96	31.11	19.06	28.98	17.00	27.02	15.07	30.02	15.77	28.64	9.32	15.89
Corr	42.38	48.67	40.86	49.58	30.18	35.35	43.22	52.57	41.65	51.98	14.23	20.96
Comp	42.54	48.09	34.02	42.05	45.28	59.00	25.01	36.72	28.96	37.23	34.99	33.52

Table 3: The results of different multi-label text classifiers on the AAPD dataset. The symbols "+" and "-" denote whether the augmentation data generated by LD-VAE is utilized during classifier training or not, respectively.

combination of these representations. *iv)* **GPT3.5**[1]: We direct GPT3.5 to produce texts based on concatenated label phrases. Moreover, we deploy few-shot in-context learning, enabling GPT3.5 to observe all existing examples labeled with the respective compositions from the label generative model. *v)* **GPT2-PT (Li and Liang 2021)**: This approach fine-tunes GPT2 using prefix-tuning, generating text that begins with the concatenated label phrases.

Main Results and Discussions

Analysis of Generators   Table 2 shows that utilizing synthetic data, generated by *all* types of generators, enhances the performance of the BERT classifier on the CG splits of three benchmarks across various evaluation metrics. This evidence underscores the effectiveness of our data augmentation approach in addressing the CG challenge within MLTC. Both LS-PT and LD-VAE outperform the other baselines, highlighting the essential role of disentangled label representations for generating high-quality instances with novel label compositions. In contrast, the Concat baseline underperforms, likely because the concatenated text is neither semantically nor syntactically coherent. Flan-T5 and GPT2-PT produce text based on label representations encoded via Transformer layers. However, we believe their encoding methods may result in entangled label representations, which may explain their inferior performance in data augmentation compared to our method. While VAE-DPrior adopts disentanglement learning and latent label representations, its lack of a label-specific representation for each label makes it less directly comparable to our approach. Even though GPT3.5 is recognized as a powerful language model, it does not excel in augmenting the CG abilities of MLTC models, potentially because it is exposed to only a few-shot examples. It's worth noting that AAPD excludes single-labeled instances, making learning disentangled label representations challenging. Yet, LD-VAE still excels on AAPD, whereas Flan-T5, despite its strong performance on the other two datasets, falls short.

Analysis of Classifiers   Table 3 shows that all MLTC models evaluated struggle with CG, each for unique reasons. Qiu et al. (2022b) found that models using parameter-efficient fine-tuning (PEFT) generally outperform merely fine-tuned ones in out-of-distribution CG scenarios in semantic parsing, a natural language understanding (NLU) task. However, BERT+P, despite employing PEFT, does not outper-

form fine-tuned baseline in MLTC — a task also under NLU. MLTC models designed to learn label correlations, like BERT+MAGNET, BERT+SGM, and T5+CLP, score better in correctness but have lower completeness scores than other baselines, suggesting they tend to predict only a subset of the ground truth. Interestingly, despite T5+CLP achieving several state-of-the-art results on current MLTC benchmarks with the standard data split, it performs the worst among all baselines. We conjecture that this line of work, despite its popularity, might produce models prone to learning spurious correlations among labels. In contrast, BERT+DBloss, designed to tackle label imbalance, leans towards over-predicting labels with its high completeness score. We also investigate the impact of synthetic data, generated by LD-VAE, on the performance of these models. Incorporating this synthetic data into training significantly boosts *all* models regarding the evaluation metrics, demonstrating the effectiveness of our data augmentation strategy in helping various MLTC models address CG challenges.

Ablation Study

Support Size   We investigate the influence of the sizes of support set on three aspects when: *i)* fine-tuning the label generator, *ii)* learning the conditional text generator, and *iii)* training the classifier. In each experiment in Table 4, we fix the support set data size for the other two aspects at 50 and only vary the data size for one aspect at a time. All experiments share the same quality control filter and test set for fair comparisons. Key takeaways include: *i)* With just 50 sampled examples, the label generator can estimate a label composition distribution reasonably close to what is achieved with 250 examples. However, a zero-shot approach that relies solely on the pre-trained knowledge of label token distribution remains challenging, resulting in the classifier accuracy being about 10% lower than using 50 examples. *ii)* Enhancing the conditional generator with additional support data has minimal impact on MLTC performance, given that even 250 examples occupy just a small fraction of the overall training data. This further solidifies our hypothesis that the conditional distribution does not shift across CG splits. *iii)* Support data size crucially affects classifier training. More human-crafted data improves classifier performance in CG.

Quality Control   We investigate the effectiveness of QC by comparing it with random selection. The sizes of selected synthetic data are equal for both settings. As shown in Table 5, our BERT-based filter improves the quality of the generated examples, as evidenced by the higher accuracy of the

---

[1]https://chat.openai.com/

Size of Support	Label Generator	Conditional Text Generator	Classifier
0	23.56	31.41	32.37
50	32.09	32.09	32.09
100	30.02	31.70	33.31
250	31.61	31.92	35.23

Table 4: Accuracies of the BERT classifier on AAPD, across three modules, with varying support data sizes, using LD-VAE as the conditional text generator.

	SemEval		AAPD	
	filter	random	filter	random
Flan-T5	8.01	6.18	26.46	24.01
GPT2-PT	6.81	6.63	28.03	27.16
LS-PT	8.29	6.93	30.21	29.32
LD-VAE	8.44	6.43	31.11	28.82

Table 5: BERT classifier accuracies with and without filtering synthetic data generated by various generators.

classifier trained on filtered data. We note that the filter tends to discard low-quality synthetic examples that are either irrelevant to the target label composition or texts with no practical meaning, such as Twitter tags and blank text.

Disentanglement Figure 2 shows entangled representations of label phrases from GPT2-PT. In contrast, the label phrase representations encoded by Flan-T5 remain invariant regardless of label composition changes, with representations of the same labels clustering closely. This may be due to Flan-T5's unique pre-training method across thousands of NLP tasks, allowing it to encounter diverse text compositions. This unique training method may explain why Flan-T5 outperforms most of the baselines. On the other hand, our methods disentangle latent representations more effectively than both GPT2-PT and Flan-T5. Notably, LD-VAE samples representations from a more continuous space rather than focusing on a singular point for each label, resulting in a more cohesive and fluent generated text than LS-PT, given our manual inspection. A further experiment reveals that replacing the label-conditioned priors in our LD-VAE with a normal distribution, as seen in vanilla VAEs, leads to a 5% drop in BERT classifier accuracy on AAPD. This emphasizes the significance of disentanglement learning.

Related Work

Multi-Label Text Classification In the field of MLTC, studies address the critical challenge of label correlation through methods such as incorporating label co-occurrence (Pal, Selvakumar, and Sankarasubbu 2020; Liu, Yuan, and Wang 2020) and employing correlation loss functions (Chai et al. 2022; Alhuzali and Ananiadou 2021). Some studies also adopt sequence-to-sequence approaches for MLTC, wherein the decoder takes label correlations into account (Yang et al. 2018; Huang et al. 2021a). Beyond label correlation, several works employ attention mechanisms to incorporate contextual label information during the pre-
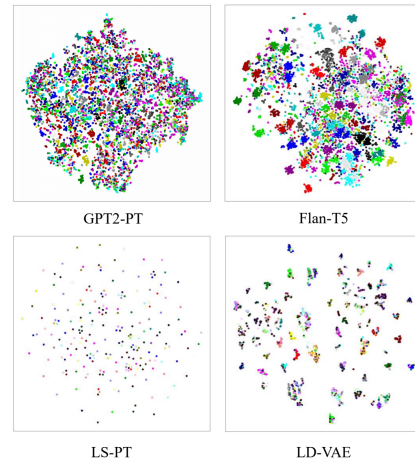

Figure 2: T-SNE visualization of Transformer-encoded label phrase representations from GPT2-PT and Flan-T5 versus latent label representations in the prefixes of LS-PT and LD-VAE. Each label, within varying label compositions from the training set $\mathcal{D}_{cg}$ of AAPD, is represented by a distinct colour.

diction (Xiao et al. 2019; Huang et al. 2021c). Additionally, various works address the challenge of label distribution imbalance in MLTC (Huang et al. 2021d; Yang et al. 2020; Cao et al. 2019). However, these studies mainly deal with the scarcity of individual labels. In contrast, our focus is on the datasets where individual labels are well-represented, but certain label combinations remain sparse.

Compositional Generalization CG has been explored in various NLP domains, including semantic parsing (Qiu et al. 2022a; Andreas 2020; Yang, Zhang, and Yang 2022; Qiu et al. 2022b; Haroutunian et al. 2023), controllable text generation (Li et al. 2022; Zeng et al. 2023), single-label classification (Li et al. 2022), and machine translation (Li et al. 2021a; Russin et al. 2019; Zheng and Lapata 2022). Typically, these studies enhance the CG capabilities of models in their respective tasks using methods such as data augmentation (Jia and Liang 2016; Andreas 2020; Qiu et al. 2022a), leveraging pre-trained knowledge from language models (Qiu et al. 2022b; Furrer et al. 2020), employing disentanglement learning (Zheng and Lapata 2022; Montero et al. 2020) for improved latent representations, or a hybrid approach (Li et al. 2022), similar to ours.

Conclusion

In summary, we examined the CG challenges in current MLTC models using our unique evaluation metrics and data splits. Our findings reveal a significant deficit in their CG capabilities, limiting their generalization to rare compositional concepts. To address this, we introduced a data augmentation method paired with two conditional text generators that learn disentangled label representations, enabling higher-quality text generation. Empirical results demonstrate that our method significantly mitigates the CG issue for MLTC models, with our generators surpassing other baseline counterparts in enhancing CG capabilities of these models.

Acknowledgments

References

Alhuzali, H.; and Ananiadou, S. 2021. SpanEmo: Casting Multi-label Emotion Classification as Span-prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 1573–1584. Online: Association for Computational Linguistics.

Andreas, J. 2020. Good-Enough Compositional Data Augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7556–7566.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

Chai, Y.; Teng, C.; Fei, H.; Wu, S.; Li, J.; Cheng, M.; Ji, D.; and Li, F. 2022. Prompt-Based Generative Multi-label Emotion Prediction with Label Contrastive Learning. In *Natural Language Processing and Chinese Computing*, 551–563.

Cho, K.; Merrienboer, B.; Gulcehre, C.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*.

Chomsky, N. 2014. *Aspects of the Theory of Syntax*. 11. MIT press.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling Instruction-Finetuned Language Models.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the NAACL*, 4171–4186.

Finegan-Dollak, C.; Kummerfeld, J. K.; Zhang, L.; Ramanathan, K.; Sadasivam, S.; Zhang, R.; and Radev, D. 2018. Improving Text-to-SQL Evaluation Methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 351–360.

Furrer, D.; van Zee, M.; Scales, N.; and Schärli, N. 2020. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv preprint arXiv:2007.08970*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Haroutunian, L.; Li, Z.; Galescu, L.; Cohen, P.; Tumuluri, R.; and Haffari, G. 2023. Reranking for Natural Language Generation from Logical Forms: A Study based on Large Language Models. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*.

Huang, C.; Trabelsi, A.; Qin, X.; Farruque, N.; Mou, L.; and Zaïane, O. 2021a. Seq2Emo: A Sequence to Multi-Label Emotion Classification Model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4717–4724. Online: Association for Computational Linguistics.

Huang, S.; Li, Z.; Qu, L.; and Pan, L. 2021b. On Robustness of Neural Semantic Parsers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 3333–3342.

Huang, X.; Chen, B.; Xiao, L.; Yu, J.; and Jing, L. 2021c. Label-aware document representation via hybrid attention for extreme multi-label text classification. *Neural Processing Letters*, 1–17.

Huang, Y.; Giledereli, B.; Köksal, A.; Özgür, A.; and Ozkirimli, E. 2021d. Balancing Methods for Multi-label Text Classification with Long-Tailed Class Distribution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 8153–8161. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Jia, R.; and Liang, P. 2016. Data Recombination for Neural Semantic Parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12–22.

Keysers, D.; Schärli, N.; Scales, N.; Buisman, H.; Furrer, D.; Kashubin, S.; Momchev, N.; Sinopalnikov, D.; Stafiniak, L.; Tihon, T.; et al. 2019. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. In *International Conference on Learning Representations*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lee, D.; Yoon, J.; Song, J.; Lee, S.; and Yoon, S. 2019. One-shot learning for text-to-sql generation. *arXiv preprint arXiv:1905.11499*.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4582–4597.

Li, Y.; Yin, Y.; Chen, Y.; and Zhang, Y. 2021a. On Compositional Generalization of Neural Machine Translation. 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)*, 4767–4780.

Li, Z.; Qu, L.; Huang, S.; and Haffari, G. 2021b. Few-Shot Semantic Parsing for New Predicates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 1281–1291.

Li, Z.; Qu, L.; Xu, Q.; Wu, T.; Zhan, T.; and Haffari, G. 2022. Variational Autoencoder with Disentanglement Priors for Low-Resource Task-Specific Natural Language Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 10335–

10356. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Liu, H.; Yuan, C.; and Wang, X. 2020. Label-Wise Document Pre-training for Multi-label Text Classification. In Zhu, X.; Zhang, M.; Hong, Y.; and He, R., eds., *Natural Language Processing and Chinese Computing*, 641–653. Cham: Springer International Publishing.

Liu, X.; Ji, K.; Fu, Y.; Tam, W.; Du, Z.; Yang, Z.; and Tang, J. 2022. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 61–68. Dublin, Ireland: Association for Computational Linguistics.

Maiya, S. 2019. Movie-Genre-Multi-Label-Text-Classification. https://github.com/shashankvmaiya/Movie-Genre-Multi-Label-Text-Classification.

Mohammad, S.; Bravo-Marquez, F.; Salameh, M.; and Kiritchenko, S. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the SemEval*, 1–17.

Montero, M. L.; Ludwig, C. J.; Costa, R. P.; Malhotra, G.; and Bowers, J. 2020. The role of disentanglement in generalisation. In *International Conference on Learning Representations*.

Pal, A.; Selvakumar, M.; and Sankarasubbu, M. 2020. MAGNET: Multi-Label Text Classification using Attention-based Graph Neural Network. In Rocha, A. P.; Steels, L.; and van den Herik, H. J., eds., *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, Valletta, Malta, February 22-24, 2020*, 494–505. SCITEPRESS.

Qiu, L.; Shaw, P.; Pasupat, P.; Nowak, P.; Linzen, T.; Sha, F.; and Toutanova, K. 2022a. Improving Compositional Generalization with Latent Structure and Data Augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4341–4362.

Qiu, L.; Shaw, P.; Pasupat, P.; Shi, T.; Herzig, J.; Pitler, E.; Sha, F.; and Toutanova, K. 2022b. Evaluating the Impact of Model Scale for Compositional Generalization in Semantic Parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 9157–9179. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Russin, J.; Jo, J.; O'Reilly, R. C.; and Bengio, Y. 2019. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv preprint arXiv:1904.09708*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Z.; and Culotta, A. 2021. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14024–14031.

Xiao, L.; Huang, X.; Chen, B.; and Jing, L. 2019. Label-Specific Document Representation for Multi-Label Text Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 466–475. Hong Kong, China: Association for Computational Linguistics.

Yang, J.; Zhang, L.; and Yang, D. 2022. SUBS: Subtree Substitution for Compositional Semantic Parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 169–174.

Yang, P.; Sun, X.; Li, W.; Ma, S.; Wu, W.; and Wang, H. 2018. SGM: Sequence Generation Model for Multi-label Classification. In *Proceedings of the COLING*, 3915–3926.

Yang, W.; Li, J.; Fukumoto, F.; and Ye, Y. 2020. HSCNN: A Hybrid-Siamese Convolutional Neural Network for Extremely Imbalanced Multi-label Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6716–6722. Online: Association for Computational Linguistics.

Yarullin, R.; and Serdyukov, P. 2021. BERT for Sequence-to-Sequence Multi-label Text Classification. In van der Aalst, W. M. P.; Batagelj, V.; Ignatov, D. I.; Khachay, M.; Koltsova, O.; Kutuzov, A.; Kuznetsov, S. O.; Lomazova, I. A.; Loukachevitch, N.; Napoli, A.; Panchenko, A.; Pardalos, P. M.; Pelillo, M.; Savchenko, A. V.; and Tutubalina, E., eds., *Analysis of Images, Social Networks and Texts*, 187–198. Cham: Springer International Publishing.

Zeng, W.; Zhao, L.; He, K.; Geng, R.; Wang, J.; Wu, W.; and Xu, W. 2023. Seen to Unseen: Exploring Compositional Generalization of Multi-Attribute Controllable Dialogue Generation. *arXiv preprint arXiv:2306.10317*.

Zheng, H.; and Lapata, M. 2022. Disentangled Sequence to Sequence Learning for Compositional Generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4256–4268.