Big GCVAE: decision-making with adaptive transformer model for failure root cause analysis in semiconductor industry

Kenneth Ezukwoke, Anis Hoayek, Mireille Batton-Hubert, Xavier Boucher, Pascal Gounet, Jérôme Adrian

Big GCVAE: Decision-Making with Adaptive Transformer Model for Failure Root Cause Analysis in Semiconductor Industry

Kenneth Ezukwoke[*1], **Anis Hoayek**[1], **Mireille Batton-Hubert**[1], **Xavier Boucher**[2], **Pascal Gounet**[3], **Jérôme Adrian**[3]

[1]**Mines Saint-Étienne, Institute Henri Fayol, Mathematics and Industrial Engineering Dept.,**
Univ. Clermont Auvergne, CNRS UMR 6158 LIMOS, Saint-Etienne, France
[2]**Mines Saint-Étienne, CIS- Center for Biomedical Engineering**,
Univ. Clermont Auvergne CNRS UMR 6158 LIMOS, Saint-Etienne, France
[3]**Failure Analysis Laboratory, STMicroelectronics**, Grenoble, France

Abstract

Pre-trained large language models (LLMs) have gained significant attention in the field of natural language processing (NLP), especially for the task of text summarization, generation, and question answering. The success of LMs can be attributed to the attention mechanism introduced in Transformer models, which have outperformed traditional recurrent neural network models (e.g., LSTM) in modeling sequential data. In this paper, we leverage pre-trained causal language models for the downstream task of failure analysis triplet generation (FATG), which involves generating a sequence of failure analysis decision steps for identifying failure root causes in the semiconductor industry. In particular, we conduct extensive comparative analysis of various transformer models for the FATG task and find that the BERT-GPT-2 Transformer (Big GCVAE), fine-tuned on a proposed Generalized-Controllable Variational AutoEncoder loss (GCVAE), exhibits superior performance in generating informative latent space by promoting disentanglement of latent factors. Specifically, we observe that fine-tuning the Transformer style BERT-GPT2 on the GCVAE loss yields optimal representation by reducing the trade-off between reconstruction loss and KL-divergence, promoting meaningful, diverse and coherent FATs similar to expert expectations.

1 Introduction

Failure Analysis Triplet Generation (FATG) is a scientific process that aims to generate a sequence of failure analysis texts for a given failure description. We approach FATG as a data-to-text generation task, where the input is a Failure Description Report (FDR) represented as structured tabular data, and the output is a lengthy sequence of failure analysis triplets. Data-to-text generation is the automatic generation of natural language reports from non-contextual, non-linguistic inputs (Gatt and Krahmer, 2018). Early approaches employed knowledge-based expert systems (Kukich, 1983) to generate natural language reports from computer databases. In recent times however, there has been a notable rise in the use of end-to-end auto-encoding architectures (Bahdanau et al., 2015) for the purpose of data-to-text generation (natural language inference). These architectures involve training encoder-decoder frameworks within the framework of deep learning-based sequence-to-sequence (Seq2Seq) models. Two prominent types of Seq2Seq models, namely Long-Short-Term Memory (LSTM) (Liu et al., 2016; Prajit Ramachandran, 2016; McCann et al., 2017) and Transformers (Vaswani et al., 2017), have garnered significant attention in the field of natural language generation (NLG).

The LSTM-based Seq2Seq models have proven to be effective in capturing the dependencies and contextual information present in sequential data, making them well-suited for NLG tasks. By employing an encoder-decoder architecture, the LSTM models are capable of encoding the input data and generating coherent and contextually relevant text as output. Transformers, on the other hand, are a type of neural network architecture that emerge as powerful alternatives to LSTM models for sequence modeling tasks. Transformers use a self-attention mechanism, which allows them to capture global dependencies in the input sequence. This makes Transformers particularly suitable for natural language generation (NLG) tasks, which require a broader context and can benefit from capturing complex relationships between words or entities. In the context of failure analysis triplet generation (FATG), the attention mechanism in Transformers is employed to generate long-sequenced, non-contextual natural language decisions made by experts during failure root cause analysis. This means that we can use Transformers to generate textual response for analyzing failed components,

[*] Corresponding author: kennedyczar@gmail.com

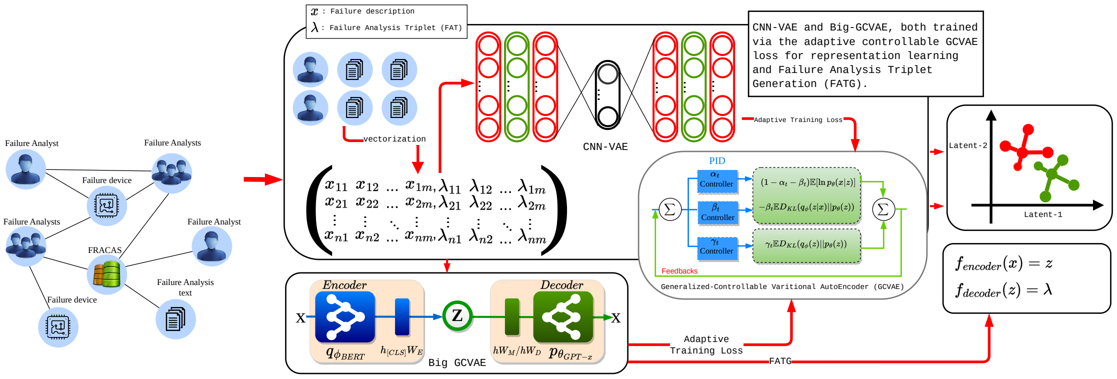

Failure Analyst
Failure Analysts
Failure device
Failure Analysts
Failure Analyst
FRACAS
Failure Analysis text
Failure device
x : Failure description
λ : Failure Analysis Triplet (FAT)
vectorization
CNN-VAE
CNN-VAE and Big-GCVAE, both trained via the adaptive controllable GCVAE loss for representation learning and Failure Analysis Triplet Generation (FATG).
Adaptive Training Loss
PID
α_t Controller
β_t Controller
γ_t Controller
$(1-\alpha_t-\beta_t)\mathbb{E}[\ln p_\theta(x|z)]$
$-\beta_t \mathbb{E} D_{KL}(q_\phi(z|x)||p_\theta(z))$
$\gamma_t \mathbb{E} D_{KL}(q_\phi(z)||p_\theta(z))$
Feedbacks
Generalized-Controllable Varitional AutoEncoder (GCVAE)
Latent-2
Latent-1
$f_{encoder}(x) = z$
$f_{decoder}(z) = \lambda$
Encoder
Decoder
X
Z
X
$q_{\phi_{BERT}}$
$h_{[CLS]}W_E$
hW_M/hW_D
$p_{\theta_{GPT-x}}$
Big GCVAE
Adaptive Training Loss
FATG

Figure 1: Symbolic representation of FATG decision-making process

even if the input data is not in a contextual format. The adoption of end-to-end auto-encoding architectures, coupled with the training of encoder-decoder frameworks employing LSTM or Transformer-based Seq2Seq models, has significantly advanced the domain of data-to-text generation. These methodologies have showcased remarkable effectiveness in a numerous of Natural Language Generation (NLG) tasks, thereby fostering progress in machine translation, text summarization, question answering, and natural language inference.

The framework introduced in this paper for variational auto-encoder fine-tuning methodology includes the following aspects:

- Robust fine-tuning of a coupled BERT-encoder and GPT-decoder: This approach involves leveraging the Generalized Controllable Variational AutoEncoder (GCVAE) loss with adaptive hyperparameters. These adaptive hyperparameters are instrumental in balancing the tradeoff between the quality of text reconstruction and the disentanglement of latent factors in the representation space, leading to structured human-like understanding and generation (See Table 3). We also benefit from maximizing the informativeness of the latent space when finetuning using the GCVAE loss for the domain task of failure analysis generation.

- Maximizing efficiency of GCVAE loss for optimal latent space representation: To optimize the latent space representation, this methodology deliberately avoids the conventional practice of masking the BERT encoder. Instead, it permits the unimpeded flow of information through the bottleneck. During this process, the primary focus is placed on evaluating the mutual information during both inference and reconstruction when optimizing of this loss function.

The remaining section of the paper follows with related works in Section (2), failure reporting and corrective action system in Section (3), Pretrained Large Language Models, in Section (4), Extensive experimentation with quantitative and qualitative analysis in Section (5) with conclusion and limitations in Sections (6) and (7) respectively.

2 Related work

NLP in manufacturing. NLP plays a vital role in improving quality control processes in the manufacturing industry. By analyzing textual data from various sources, such as customer feedback, sensor logs, and maintenance reports, NLP algorithms can identify patterns and anomalies related to product defects. This enables manufacturers to detect and address quality issues promptly, thereby enhancing product reliability and customer satisfaction. Notable research in this domain include works by Biffl and Halling (2003); Loniewski et al. (2010); Binkhonain and Zhao (2019) show remarkable progress of NLP for defect detection. By leveraging sentiment analysis and topic modeling, manufacturers can gain real-time visibility into market trends, customer preferences, and demand patterns (Garg et al., 2021; Biswas et al., 2022; Toorajipour et al., 2021; Pournader et al., 2021) for Supply Chain Management and Demand Forecasting. This paper, along with previous publications by Ezukwoke et al. (2021); Wang et al. (2022); Rammal et al. (2023a), represents pioneering efforts in exploring the application of Natural Language Processing (NLP) and machine/deep learning techniques for the identification of failure root cause analysis in semiconductor industry.

Intelligent decision-making necessitates the intelligent representation of historical expert decisions, particularly in cases where a single type of failure can have multiple failure analysis approaches. Hence, to design intelligent systems capable of handling such scenarios, it is essential to extract meaningful information from failure reporting systems. This extraction process enables the creation of intelligent systems that can effectively interpret and utilize the knowledge derived from historical expert decisions, leading to improved decision-making capabilities in failure analysis. Given the huge amount of textual data obtained from the FA final report therefore, NLP as a subdomain of artificial intelligence is used to structure and find decision patterns in such data. Notable works by Zimmer et al. (2019) used NLP techniques to find pattern of decisions made by expert during RAMP-UP production. Yue in Yue et al. (2018) used CNN-LSTM for industrial fault diagnosis and prognosis.

Representation learning and language modeling. To address the challenge of representation of unstructured text obtained from industrial data, it is crucial to explore techniques that can reduce the dimensionality of the vector space, resulting from numerization, while preserving the contextual meaning of the text. AutoEncoders (Oshri, 2015; Chen and Zaki, 2017) and Variational Autoencoders (Dirichlet version) (Xiao et al., 2018) have been widely employed for this purpose. Recent studies have shown that Convolutional Neural Networks based on Variational Autoencoders (Liu et al., 2020) outperform other methods in terms of preserving semantic meaning in large-scale text data. Ezukwoke et al. (2021) used β-Variational AutoEncoder while Rammal et al. (2023b) used Variational AutoEncoders together with genetic algorithm to find a well disentangled optimal representation space for failure analysis dataset where clusters of decisions can be found using K-means or Gaussian mixture model. This is attributed to their ability of VAEs to extract semantic representations from textual data.

The challenge with using Variational AutoEncoder for modeling complex data is the problem of *vanishing* Kullback–Leibler (KL) divergence (Bowman et al., 2016), where the unit Gaussian prior of the encoder matches the posterior. This results in a non-informative latent z, due to lack of disentanglement (that is, collapsing different factors of variation into a small region of the latent space). Attempts to address the *vanishing* KL problem includes using KL annealing schemes (Bowman et al., 2016; Higgins et al., 2017; Fu et al., 2019a), importance weighted autoencoders (Burda et al., 2016), conditional variational autoencoder (Zhao et al., 2017), controllable variational autoencoder (Shao et al., 2020) and generalized controllable variational autoencoder (Ezukwoke et al., 2022a). Transformer-based Variational Autoencoders (Liu and Liu, 2019) have demonstrated improved sentence generation with more meaningful content and coherent semantics in the latent space. Further research reveal that by pretraining Transformers with a BERT-encoder and a GPT2-decoder on variational loss (Li et al., 2020), the inherent solution to the vanishing KL issue was achieved, leading to the state-of-the-art performance in inference tasks.

Our paper is the first to consider fine-tuning a pretrained Transformer (BERT-encoder and a GPT2-decoder) on a robust generalized controllable variational autoencoder (GCVAE) loss (Ezukwoke et al., 2022a) and also applying it to the semiconductor manufacturing industry for decision inference for failure root cause analysis. The results (in Sections 5.5 and 5.6) suggests that the GCVAE loss supports disentanglement and is capable of reducing the trade-off between the reconstruction and regularization, leading to more informative latent space.

3 Failure Reporting and Corrective Action System: FRACAS

FRACAS stands for "Failure Reporting, Analysis, and Corrective Action System." It is a systematic approach used in industries, particularly in engineering and manufacturing, to manage the identification, analysis, and resolution of failures or defects in products, systems, or processes. The FRACAS process involves the collection of failure data, analysis of the root causes of failures, and the implementation of corrective actions to prevent recurrence.

At the core of FRACAS is its database management system (DBMS), designed to categorize failure modes into critical categories, enabling the identification of product life cycle processes that require significant attention for enhancing reliability (Mario, 1992). Reports generated from the FRACAS DBMS provide valuable insights into failure types, origins of detection, and a series of analyses represented as triplets, consisting of step type, substep technique, and equipment, proposed to identify the root cause of failures. These reports also present conclusions drawn from the failure analysis.

Throughout the subsequent sections of this paper, the term **pre-triplet** will refer to failure descriptions, while **triplets** (Step type; Substep technique; Equipment) will denote the collection of analyses, each comprising three key components. Each triplet contributes to a failure decision, and the main objective of this paper is to generate a set of n triplets that correspond to a specific failure description.

3.1 FRACAS Variables

We describe the three important variables (the triplets) for decision-making as follows:

- **Step type**:The initial stage of fault analysis entails determining the fault analysis step type, which represents the first analysis proposed by the expert once the sample is validated for analysis. At STMicroelectronics, the fault analysis process is predominantly composed of Non-destructive Inspection, Electrical Failure Verification, Sample Preparation, Physical Analysis, Global Fault Localization, and other step types, collectively accounting for 94.3% of all conducted fault analyses. There is often a correlation between the requested activity and the specific analysis conducted. The expertise of an expert plays a crucial role in comprehending the rationale behind a specific request and selecting the appropriate path to identify the origin of the defects.

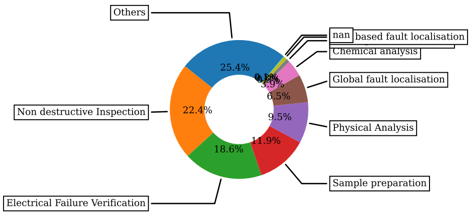


Figure 2: Percentage distribution of Step types.

The Figure (2) shows the percentage distribution of Step type taken during failure analysis between 2019-2021.

- **Substep technique**: In the context of fault analysis, each step type is accompanied by a corresponding substep technique. A comprehensive set of 91 distinct substep techniques is employed for sub-analysis purposes. Among these techniques, certain substep techniques are frequently associated with specific step types. Notably, package decap, optical microscopy, SAM (Scanning Acoustic Microscopy), X-ray, SEM (Scanning Electron Microscopy), die delaying, FIB (Focused Ion Beam) cross-section, continuity test, electrical parametric test, and mechanical cross-section are the most commonly used substep techniques.

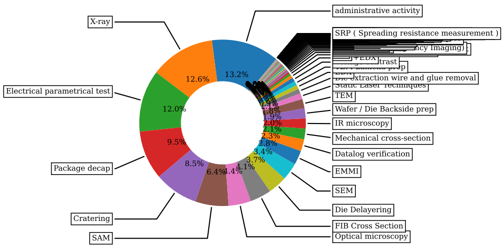


Figure 3: Percentage distribution of Substep tehniques.

To visualize the distribution of the various substep techniques in the dataset, a pie chart is presented in Figure (3). This chart offers valuable insights into the relative frequency of each substep technique within the fault analysis dataset.

- **Equipment**: Equipment are the tools used for failure analysis. The equipment are 1348 equipment types found in the data set. OM113-LEICA M165C, PK103-PHOENIX X-RAY NANOMEX, ZZ003 - CRI7, ZZ003 - CRI6, MICROSCOPE LEICA DM2700M, SAM, AGR XRAY01, AGR MICROXCT 200 X-RADIA 3D, AGR STEREOSCOPIC MICROSCOPE M8-FA, AGR XPREP01, AGR DUALBEAM HELIOS 400, AGR PARALLEL LAPPER 3-FA are the top ten equipments frequently used for analysis (See Figure (4)).

3.2 Formalization

. In the context of FATG, we propose a general symbolic representation for the decision-making process (See Figure 1). In subsequent sections, we explore language models, specifically pre-trained

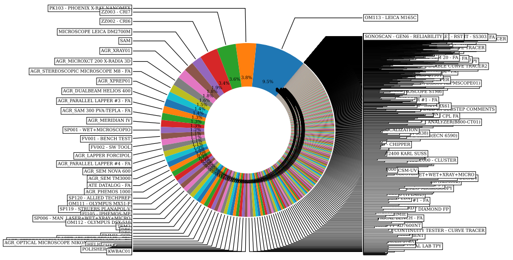


Figure 4: Percentage distribution of Equipment.

language models based on the Transformer architecture, for FATG. To assess the performance of different models, we employ widely used scoring metrics such as BLEU, ROUGE, and METEOR. We initially formalize our problem within a probabilistic graphical framework and subsequently extend it to language modeling.

Given failure analysis description (FDR) $\{x_i\}_{i=1}^N \in \mathbb{R}^D$, where N is the number of observation and D is the dimension of the preprocessed data, and a set of failure analysis triplets $\{\lambda_i\}_{i=1}^N \in \mathbb{R}^M$, where M is the dimension of λ. We express the FATG as data-to-text problem by defining the input space as a joint space of x and λ. Let us begin by modeling them individually, the FDR (input space) probability mass function is,

$$p_x(x) = \prod_{i=1}^{N} p_x(x_i|x_{1:i-1}) \tag{1}$$

and the failure analysis triplets (FAT),

$$p_\lambda(\lambda) = \prod_{i=1}^{N} p_\lambda(\lambda_i|\lambda_{1:i-1}) \tag{2}$$

The joint probability space is given as,

$$p_{x,\lambda}(x,\lambda) = \prod_{i=1}^{N} p_{x,\lambda}(x_i,\lambda_i|x_{1:i-1},\lambda_{1:i-1}) \tag{3}$$

compact representation. We present a compact representation for Equation (3) as follows. Let us represent the joint space between failure description $x \in \mathbb{R}^D$ and failure analysis triplets $\lambda \in \mathbb{R}^M$ i.e. $\{x_i, \lambda_i\}_{i=1}^N \in \mathbb{R}^K$ as $\Lambda \in \mathbb{R}^K$, where $K = D + M$ is the dimension of the joint space- a sum of the dimensions of x and λ respectively. We define the compact joint probability space between x and λ as,

$$p(\Lambda) = \prod_{i=1}^{N} p(\Lambda_i|\Lambda_{1:i-1}) \tag{4}$$

This compact joint space can be modeled into a likelihood function, $\sum_{i=1}^N \log p(\Lambda_i|\Lambda_{1:i-1};\phi)$ and its parameter, ϕ estimated using a deep neural network.

4 Pretrained Large Language Model

4.1 Generative Pre-trained Transformer

Generative Pre-trained Transformer model (GPT) (Radford and Narasimhan, 2018) is a type of transformer model that uses a multi-layer Transformer decoder (Liu et al., 2018) instead of the encoder-decoder model introduced by (Vaswani et al., 2017). GPT is known for its ability to generate coherent and contextually relevant text including human-like responses, sentences completion, and even writing entire articles. By leveraging its pre-trained knowledge and fine-tuning on domain specific FA task, GPT is capable of understanding the semantic and syntactic structures of failure analysis description and triplets, making it suitable for failure triplet generation.

The model structure begins with training in an unsupervised setting corpus of tokens, $\mathcal{U} = \{u_1, \ldots, u_n\}$ with a standard likelihood objective to maximize:

$$L_1(\mathcal{U}) = \sum_i logP(u_i|u_{i-k}, \ldots, u_{i-1}; \Theta) \tag{5}$$

Where k is the context window and Θ is the neural network parameters obtained when modeling conditional probability P. This model applies a multi-headed self-attention operation over the input context tokens followed by position-wise feed-forward layers to produce an output distribution over target tokens as follows:

$$h_0 = UW_e + W_p \tag{6}$$

$$h_i = \texttt{transformer\_block}(h_{i-1}) \quad \forall i \in [1, n] \tag{7}$$

$$P(u) = \texttt{softmax}(h_n W_e^T) \tag{8}$$

Where $U = (u_{-k}, \ldots, u_{-1})$ is the context vector of tokens, n here is the number of layers while W_e and W_p are the token embedding and position embedding matrix respectively.

4.2 Bidirectional Encoder Representations from Transformers

Bidirectional Encoder Representations from Transformers or BERT, is a natural language processing

(NLP) model developed to comprehensively understand the context of words in a sentence by processing language bidirectionally, capturing both left and right context. This bidirectional approach significantly enhances its grasp of semantics and meaning. BERT is pretrained on vast amounts of text data, allowing it to learn intricate language patterns and nuances. It uses the transformer architecture, known for its efficacy in modeling sequential data, and employs self-attention mechanisms to capture long-range dependencies. After pretraining, BERT can be fine-tuned for specific NLP tasks, achieving state-of-the-art results in applications such as text classification, question-answering, and sentiment analysis. During BERT's pretraining phase, it employs two main objectives: the **Masked Language Model** (**MLM**) and **Next Sentence Prediction** (**NSP**). In the MLM objective, a fraction of input tokens is randomly masked, and BERT learns to predict the original tokens in their context, using cross-entropy loss. The loss function for the MLM objective is the sum of cross-entropy losses for all the masked tokens in a training sample and is expressed as follows:

$$L_{\text{MLM}} = -\sum_{i} q_i log p_i \tag{9}$$

Where p is the predicted probability distribution over the vocabulary for the i-th masked token, and q is the encoding of the true token for the i-th masked token. It is important to note that during training, BERT typically masks only a fraction of the tokens in each input, so the sum is taken over the masked tokens. The NSP objective helps BERT understand sentence relationships by predicting whether a second sentence follows the first. These two objectives grant BERT a broad range of language comprehension skills. After pretraining on a large text corpus, BERT is fine-tuned for specific NLP tasks using task-specific loss functions, customized to meet each task's requirements. The loss function for the NSP objective is:

$$L_{\text{NSP}} = -(q log p) - (1 - q) log(1 - p) \tag{10}$$

Here p and q are the predicted probability of the next token and true label respectively.These two objectives, the MLM and NSP objectives, collectively provide BERT with a diverse set of language understanding capabilities during pretraining.

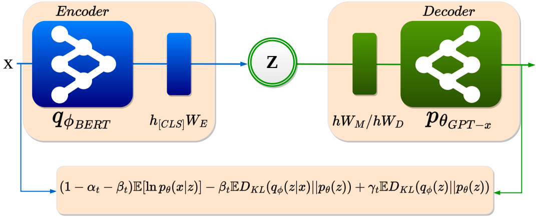

Encoder
x
$q_{\phi_{BERT}}$
$h_{[CLS]}W_E$
Z
Decoder
hW_M/hW_D
$p_{\theta_{GPT-x}}$
x
$(1 - \alpha_t - \beta_t)\mathbb{E}[\ln p_\theta(x|z)] - \beta_t \mathbb{E} D_{KL}(q_\phi(z|x)||p_\theta(z)) + \gamma_t \mathbb{E} D_{KL}(q_\phi(z)||p_\theta(z))$

Figure 5: An illustration of the Big GCVAE architecture. On the leftmost is the unmasked BERT weights loaded to the Encoder and GPT-2 weights loaded to the Decoder.

4.3 Big GCVAE

Leveraging our previous understanding of variational autoencoder for learning high-quality latent representations and optimal reconstruction of objects including text and images, we propose an improved variational Large Language Model accordingly. This model, **Big GCVAE**, is adopted for the FATG task by tying together two different transformers architectures (Encoder and Decoder-only) and fine-tuning them using GCVAE loss function (the implementation code is available on GitHub[1]). The model is structured like a classic Transformer model but loaded with pretrained weights. The Encoder is an unmasked BERT model while the decoder is a GPT-2 (for example, GPT-2 base, small or large) and fine-tuned on a loss function with adaptive hyperparameters (see Figure 5). The proposed model consists of two essential components: a generation module and an inference module, which facilitate a bidirectional mapping between the smooth continuous latent and the symbolic space, following Li et al. (2020). The generation module allows for the generation of samples from the latent space, enabling the synthesis of new instances in the symbolic domain. Conversely, the inference module enables the mapping of symbolic inputs to their corresponding latent representations, facilitating the extraction of meaningful latent features.

Inference is performed on the encoder section similar to the work of Li et al. (2020), except, rather than using the classical Evidence Lower Bound, we maximize instead the mutual information between the data x and latent z, $I_q(z, x)$ through a bottleneck, which yields a new evidence lower bound:

[1] https://github.com/FA4-0/Big-GCVAE

$$
\begin{aligned}
\mathcal{L}(\theta, \phi, \alpha, \beta, \gamma) =&(1-\alpha_t-\beta_t) \underset{z \sim q_\phi(z|x)}{\mathbb{E}}[\ln p_\theta(x|z)] \\
&- \beta_t \underset{p_{\mathcal{D}}}{\mathbb{E}} D_{KL}(q_\phi(z|x)||p_\theta(z)) \\
&+ \gamma_t D_{KL}(q_\phi(z)||p_\theta(z)) \qquad (11)
\end{aligned}
$$

We derive this lower bound in the next subsection.

4.3.1 Notations and definitions

Given a d-dimensional input space $\{x_i\}_{i=1}^N \in \mathcal{X}$ consisting of N-independently and identically distributed (**i.i.d**) samples; k-dimensional latent space $\{z_i\}_{i=1}^N \in \mathcal{Z}$ (where $k \ll d$) over which a generative model is defined. We assume an empirical prior distribution $p_\theta(z) \sim \mathcal{N}(0, I)$ to infer an approximate posterior distribution $q_\phi(z|x) \sim \mathcal{N}(z|\mu_\phi(x), \sigma^2_\phi(x)I)$, with mean $\mu_\phi(x)$ and variance $\sigma^2_\phi(x)I$ used for re-parameterization sampling of the latent space z (Kingma and Welling, 2014). We model the data using conditional distribution $p_\theta(x|z) \sim \mathcal{N}(x|\mu_\theta(x), \sigma^2_\theta(x)I)$. Let us suppose that the underlying distribution of the input space $p(x)$ follows a normal distribution, and its empirical distribution is denoted by $p_{\mathcal{D}}(x)$. $I_p(x', z)$ is the joint mutual information space between x' and z generated from the posterior $p_\theta(x|z)$ after obtaining an inference posterior $q_\phi(z|x)$.

The optimization framework proposed in (Ezukwoke et al., 2022a) is given as follows:

$$
\begin{aligned}
&\underset{\theta,\phi,\xi^+,\xi^-,\xi_p \in \mathbb{R}}{max} \quad I_p(x', z) \\
&s.t \quad \underset{p_{\mathcal{D}}}{\mathbb{E}} D_{KL}(q_\phi(z|x) \parallel p_\theta(z)) + I_p(x', z) \leq \xi^- \\
&s.t \quad - \underset{p_{\mathcal{D}}}{\mathbb{E}} D_{KL}(q_\phi(z)||p_\theta(z)) \leq \xi^+ \\
&s.t \quad I_p(x', z) \leq \xi_p \\
&s.t \quad \xi_i^+, \xi_i^-, \xi_{ip} \geq 0, \quad \forall i = 1, \ldots, n \qquad (12)
\end{aligned}
$$

The expansion of the above equations using sets of Lagrangian multipliers is as follows,

$$
\begin{aligned}
&\mathcal{L}(\theta,\phi, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-, \boldsymbol{\xi_p}, \alpha, \beta, \gamma, \boldsymbol{\eta}, \boldsymbol{\tau}, \boldsymbol{\nu}) \\
&\quad =I_p(x', z) - \beta(\underset{p_{\mathcal{D}}}{\mathbb{E}} D_{KL}(q_\phi(z|x)||p_\theta(z))+ \\
&\qquad I_p(x', z) - \xi^+) + \gamma(\underset{p_{\mathcal{D}}}{\mathbb{E}} D_{KL}(q_\phi(z)||p_\theta(z)) + \xi^-) \\
&\qquad - \alpha(I(x'; z) - \xi_p) - \boldsymbol{\eta}\xi^+ - \boldsymbol{\tau}\xi^- - \boldsymbol{\nu}\xi_p \qquad (13)
\end{aligned}
$$

The negative Lagrangian multipliers pose no challenge since they only exist to eliminate the error terms ξ^+, ξ^-, ξ_p,

$$
\begin{aligned}
&\mathcal{L}(\theta,\phi, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-, \boldsymbol{\xi_p}, \alpha, \beta, \gamma, \boldsymbol{\eta}, \boldsymbol{\tau}, \boldsymbol{\nu}) \\
&\quad =(1-\alpha-\beta)I_p(x', z) - \beta \underset{p_{\mathcal{D}}}{\mathbb{E}} D_{KL}(q_\phi(z|x)||p_\theta(z)) \\
&\qquad + \gamma D_{KL}(q_\phi(z)||p_\theta(z)) + (\beta - \boldsymbol{\eta})\xi^+ + (\gamma - \boldsymbol{\tau})\xi^- \\
&\qquad + (\alpha - \boldsymbol{\nu})\xi_p \qquad (14)
\end{aligned}
$$

We take the gradient over the loss, $\nabla\mathcal{L}$ for ξ^-, ξ^+, ξ_p and apply KKT optimality conditions to obtain,

$$
\begin{aligned}
\mathcal{L}(\theta, \phi, \alpha, \beta, \gamma) &= (1-\alpha-\beta)I_p(x', z) \\
&- \beta \underset{p_{\mathcal{D}}}{\mathbb{E}} D_{KL}(q_\phi(z|x)||p_\theta(z)) \\
&+ \gamma D_{KL}(q_\phi(z)||p_\theta(z)) \qquad (15) \\
&= (1-\alpha-\beta) \underset{z \sim q_\phi(z|x)}{\mathbb{E}}[\ln p_\theta(x|z)] \\
&- \beta \underset{p_{\mathcal{D}}}{\mathbb{E}} D_{KL}(q_\phi(z|x)||p_\theta(z)) \\
&+ \gamma D_{KL}(q_\phi(z)||p_\theta(z)) \qquad (16)
\end{aligned}
$$

We set the Lagrangian adaptive hyperparameters as follows,

$$
\begin{aligned}
\mathcal{L}(\theta, \phi, \alpha, \beta, \gamma) =&(1-\alpha_t-\beta_t) \underset{z \sim q_\phi(z|x)}{\mathbb{E}}[\ln p_\theta(x|z)] \\
&- \beta_t \underset{p_{\mathcal{D}}}{\mathbb{E}} D_{KL}(q_\phi(z|x)||p_\theta(z)) \\
&+ \gamma_t D_{KL}(q_\phi(z)||p_\theta(z)) \qquad (17)
\end{aligned}
$$

The adaptive weight α_t controls the reconstruction error while β_t ensures the posterior latent factor $q_\phi(z|x)$ does not deviate significantly from its prior $p_\theta(z)$. Varying both terms gives us better control of the degree of disentanglement and helps us to understand the parameters affecting density disentanglement. The first term of the loss in Equation 17 with weight $(1-\alpha_t-\beta_t)$ is the reconstruction loss, the second term with weight β_t is the Kullback-Leibler divergence, and the third term with weight γ_t is a distance measure. α_t, β_t and γ_t are controllable optimizable parameters based on reconstruction loss, KL-divergence and the distance measure respectively. We select the controllable parameters as proposed by (Ezukwoke et al., 2022a).

The adaptive weight α_t controls the reconstruction error while β_t ensures the posterior latent factor $q_\phi(z|x)$ does not deviate significantly from its prior $p_\theta(z)$. Both prior, $p_\theta(z)$ and posterior, $q_\phi(z|x)$ are typically modeled by a Gaussian distribution. Note that the resulting latent vector obtained on inference is re-parameterized following VAE style (Kingma and Welling, 2014). The Encoder is parameterized by pretrained BERT (Devlin et al., 2019) weights. BERT (Bidirectional Encoder Representations from Transformers) is a model designed to predict masked or hidden words within a given text. It uses a masked language modeling objective, where a certain percentage of the input tokens are randomly masked, and the model is trained to predict the masked tokens based on the context provided by the surrounding tokens which allows it to learn bidirectional representations by leveraging both the left and right context of the masked tokens. However, the proposed model uses an unmasked BERT Encoder version to allow for the free flow of information to the latent space during learning.

Generation is done through the decoder by taking a re-parameterized latent code, z from a smooth continuous latent space with prior, $p(z)$. The text sequence x is then generated by sampling from the posterior conditional distribution $p_\theta(x|z)$, which captures the conditional relationship between the latent code and the generated text sequence, and modeled as follows:

$$p(x) = \prod_{i=1}^{N} p(x_i | x_{0:i-1}, z) \tag{18}$$

Where N is the size of the generated text sequence. the prior, $p(z)$ is modeled by a Gaussian distribution.

5 Experimentation

5.1 General Setup

Experimental Setup. The experimentation is conducted on a High-Performance Computing (HPC) cluster comprising 80 cores, 2 × Intel Xeon E5-2698 v4 2.20GHz CPUs (80 cores), 512GB RAM, and 8 × Nvidia V100 32GB GPUs.

Pre-trained GPT-2. We fine-tune the medium versions of GPT-2 with 335 million parameters after downloading the pre-trained weights through the Huggingface API[2] for the purpose of failure analysis generation. The beginning of sequence token, "bos," is set to $<$ |`startoftext`| $>$, the end of sequence token, "eos," is set to $<$ |`endoftext`| $>$, and the padding token, "pad_token," is $<$ |`pad`| $>$. The batch size for training and evaluation is 1, the weight decay is 0.05, and the number of training epochs is 100. GPT-2 is trained on 40GB of WebText data, which consists of web pages from outbound links on Reddit, excluding Wikipedia pages. In our experiment, we employ a tokenization technique known as byte-level Byte Pair Encoding (BPE) to process the text data. This method divides the text into sub-word units, allowing for more effective handling of rare or out-of-vocabulary words. The result of this tokenization process is a vocabulary containing 50,257 unique tokens with the text sequences, structured as a series of 1024 consecutive tokens. This sequence length is carefully chosen to strike a balance between capturing sufficient contextual information and managing computational efficiency during training and inference.

5.2 Big GCVAE Setup

Two key technical challenges suffice during pre-training of Big VAE (Li et al., 2020) that need to be addressed for Big GCVAE when incorporating BERT and GPT-2:

- **Sentence Representation**: Since BERT and GPT-2 use different tokenization schemes, it becomes necessary to determine how to represent sentences consistently. This involves finding a compatible representation that can bridge the gap between the two models' tokenization methods.

- **Conditional Input Adaptation**: Another challenge arises when attempting to adapt a pre-trained GPT-2 model to handle arbitrary conditional inputs without requiring retraining. While previous studies have explored controllable versions of GPT-2 by providing specific control codes or tokens, it remains unclear how to effectively ground GPT-2 to arbitrary conditional inputs, where no predefined control codes or tokens are provided.

5.2.1 Tokenization

In BERT, WordPiece Embeddings (WPE) proposed by Wu et al. (2016) is used for tokenization, with a vocabulary size of 28,996 token vocabulary for the cased version. Following the BERT convention, the initial token of each sequence is consistently assigned as a distinct classification token (`[CLS]`), and final hidden state associated with this token is used as the aggregated representation of the entire sequence. This aggregate representation serves as a valuable input for downstream tasks, enabling the model to capture and leverage the contextual information of the sequence in a meaningful manner.

In GPT-2, a modified version of Byte Pair Encoding (BPE) introduced by Radford et al. (2019) is employed for tokenization, with a vocabulary size of 50,260. Each token is represented as h_{Emb} by summing the corresponding token, position, and segment embeddings. To compute the reconstruction loss, we present a sentence using both types of tokenization: WPE for the input of the encoder and BPE for the output of the decoder.

[2] https://github.com/huggingface/transformers

5.2.2 Unmasking

Masking is a concept introduced in BERT model and involves selectively hiding certain tokens within an input sequence during the pre-training phase. A percentage of the input tokens are randomly chosen for masking to enable the model learn a bidirectional representation by predicting the masked tokens based on their context (Devlin et al., 2019). This effectively gives BERT the name, Masked Language Model (MLM). Selected tokens are then replaced with special `[MASK]` tokens. Additionally, a small portion of the selected tokens are replaced with random tokens from the vocabulary to introduce further variation, making masking an effective technique in the encoder-only transformer for token prediction or classification purposes.

However, when incorporating a decoder component, such as the GPT-2 model, to complete the Big GCVAE Encoder-Decoder model, we hypothesize that the exclusive use of masking limits the model's ability to learn a quality bidirectional representation. Consequently, this restriction hampers the generalization of the latent space and considerably diminishes the mutual information within the bottleneck. This is because the loss function of the GCVAE that we are minimizing takes into account that we are reducing the mutual information in the encoder. Therefore, masking the tokens adds an extra layer of information compression. To overcome this challenge, we propose the omission of masking, allowing for constructive summarization of mutual information within the latent space, as observed in classical transformer architectures. This allows us to revert BERT to the classic left-to-right (Peters et al., 2018; Radford and Narasimhan, 2018) bidirectional Language Model (biLM) and fitted for generative architectural pairing for failure analysis triplets generation without loss of generality.

5.2.3 Latent Injection

Following a similar approach to BERT, the initial token of each sentence in Big GCVAE is a special classification token ([CLS]). The hidden state $h_{[CLS]}$ in the last layer, corresponding to this token, is extracted as the sentence-level representation. To construct the latent representation z, we employ the use of the weighted matrix $W_E \in \mathbb{R}^{R\times H}$, where z is a P-dimensional vector and W_E is the weight matrix. In order to enable the use of z in GPT-2 decoding without necessitating retraining of the weights, two schemes are considered. These schemes aim to effectively incorporate z into the GPT-2 model during the decoding process, thereby allowing for the generation of text conditioned on the latent representation without the need for extensive model retraining.

- Within the Big GCVAE framework, the latent representation z serves as an additional memory vector, denoted as h_{Mem}, which GPT-2 attends to during decoding. This is achieved by calculating h_{Mem} as the product of the weight matrix W_M and z. The resulting h_{Mem} is a vector of length L, where L represents the number of layers in GPT-2 and H denotes the length of each vector. Each element of h_{Mem} is attended to by GPT-2 in its corresponding layer.

- In addition to the memory role, z is also directly incorporated into the original embedding layer of GPT-2. This is accomplished by adding the weighted version of z, denoted as $W_D z$, to the original embedding representation h_{Mem}. The weight matrix $W_D \in \mathbb{R}^{H\times P}$, is used to transform z into a suitable dimensionality for the addition. The resulting embedding representation is $h'_{Emb} = h_{Emb} + W_D z$.

5.3 Dataset

In our experimentation, we use real failure analysis data obtained from a semiconductor industry, specifically focusing on successful failure analysis cases from the year 2019. To prepare the data for training the transformer model, we concatenate all input features, including the triplet data, along the horizontal axis (x-axis). After preprocessing, the size of the data for the year 2019 reduces to 5809 observation (or FA analysis) of which 70% (4066) is used for training and 30% (1743) is used for evaluation. The input features used for training, also referred to as **Expert features**, encompass ten distinct aspects including *Reference*, *Subject*, *Site*, *Requested activity*, *Priority level*, *High confidentiality*, *Context*, *Objectives / Work description*, *Source of failure / request*, and *Source of failure (Detailed)*, and preprocessed using NLP techniques according to Ezukwoke et al. (2021).

5.4 Evaluation metric

NLG Evaluation. Bilingual Evaluation Understudy (BLEU) is a context-free precision-based metric for evaluating the quality of text which has been machine-translated from one natural language to another (Papineni et al., 2002; Lin and Hovy, 2003) and dialog generation task (Sai et al., 2022). It is a precision-based metric that computes the n-gram overlap between the reference (original) and its hypothesis. In particular, BLEU is the ratio of the number of overlapping n-grams to the total number of n-grams in the hypothesis. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004) is a recall-based metric similar to BLEU-N in counting the n-gram matches between the hypothesis and reference. METEOR, proposed by, Banerjee and Lavie (2005) addresses the major drawback of BLEU including, its inability to account for recall and inflexible n-gram matching by proposing an F-measure with flexible n-gram matching criteria. LEvenshtein Sequential Evaluation (LESE) (Ezukwoke et al., 2022b) metric is a measure used to quantify the dissimilarity or similarity between two sequences. The LESE metric is based on the concept of edit distance, which represents the minimum number of operations required to transform one sequence into another.

Cluster Analysis. Silhouette score (Rousseeuw, 1987) measures how well each data point is assigned to its own cluster compared to how well it could be assigned to other clusters. A Silhouette score of 1 indicates that all data points are perfectly clustered, while a Silhouette score of -1 indicates that all data points are assigned to the wrong clusters. Conversely, a low or negative score suggests overlapping clusters or misclassification. The Calinski-Harabasz index (CH index) or variance ratio criterion, originally by Caliński and Harabasz (1974), measures the ratio of the between-cluster variance to the within-cluster variance. A high CH-index suggests that the clusters are well-separated and the data points within each cluster are similar to each other. The Davies-Bouldin index (Davies and Bouldin, 1979) measures the average similarity between each cluster and the cluster that is most similar to it, relative to the size of the clusters. A lower Davies-Bouldin index value indicates better-defined and more compact clusters.

5.5 Quantitative Evaluation: Big GCVAE

We conduct performance comparison between Big GCVAE and derivative models of GCVAE, such as ControlVAE and β-VAE with Annealing KL-divergence (Li et al., 2020). We adopt two versions of Big GCVAE based on the correlation measure as follows:

Big GCVAE†: $D_{KL}(q_\phi(z)||p_\theta(z)) \leftarrow$ Maximum Mean Discrepancy.

Big GCVAE‡: $D_{KL}(q_\phi(z)||p_\theta(z)) \leftarrow$ Squared Mahalanobis distance.

For the decoder component, we employed the GPT-2 medium-size model, while the encoder is an uncased-BERT. This comparison aimed to evaluate and contrast the performances of these models. For Big VAE ($\beta > 0$), a KL thresholding scheme (Li et al., 2019; Fu et al., 2019b), where $\sum_i max[\lambda, D_{KL}(q_\phi(z_i|x)||p_\theta(z_i))]$ replaces the classical KL divergence term in the VAE loss function. The metric reported in this section and the next (Section 5.6) is for one training step.

Model	Evaluation ↓ loss	Reconstruction ↓ loss	KL divergence ↗
GPT2-M	0.19	-	-
Big VAE	**1.10**	128.34	6.49
Big ControlVAE	1.18	1.10	9.85
Big GCVAE†	1.18	**1.09**	8.23
Big GCVAE‡	1.11	**1.09**	3.80

Table 1: Performance evaluation of Big GCVAE models and its derivatives. Both Big GCVAE† and Big GCVAE‡ have the lowest reconstruction loss compared to Big VAE (Li et al., 2020).

The optimal cluster size is determined using the Bayesian Information Criterion (BIC), which involves running the GMM multiple times with different predefined parameters. Note that the cluster labels obtained from clustering the latent space applies in visualizing the clusters in t-SNE Embedding.

	Latent space z		
Model	Silhouette score ↑	CH-index ↑	DB-score ↓
Big VAE	**0.26**	1697	**1.42**
Big ControlVAE	0.19	584	1.91
Big GCVAE	0.22	**1998**	2.12
Big GCVAE	0.17	590	2.05
	t-SNE Embedding		
Big VAE	**0.17**	1430	**1.38**
Big ControlVAE	0.13	1495	2.25
Big GCVAE†	**1.17**	**2071**	1.84
Big GCVAE‡	0.10	819	7.92

Table 2: Validity indices showing the results of GMM on the latent space, z and the 2-D t-SNE embedding. High **CH-index** (Caliński and Harabasz, 1974) of Big GCVAE† indicates well separated clusters.

The performance of these models was analyzed in comparison to the GPT-2 medium-size decoder-only transformer as a baseline. Big GCVAE‡ model demonstrates superior performance compared to the benchmark Big variational model across various evaluation metrics, as indicated in Table (3). When specifically applied for the FATG task, the Big GCVAE‡ model outperforms GPT-2 medium, highlighting the efficacy of controllable Lagrangian hyperparameters in achieving optimal representation and generalizing the latent space. The use of pretrained BERT-GPT-2 weights within the Big GCVAE model significantly contributes to mitigating overfitting issues and reducing the trade-off between reconstruction and KL-divergence (see Table 1). A disentangled representation is one whose latent factors are well summarized and independently factored as a vector in the latent representation space.

Big GCVAE† model yields quality latent presentation as shown in Figure 6 with well separated clusters (See CH-index in Table 2), given their low reconstruction loss and moderate KL-divergence. The moderate properties of the Big GCVAE results in a well factored latent space whose clusters are nicely knitted in the t-SNE embedding space as shown in Figure 6. We hypothesize the reason for the fuzziness of the latent space of Big ControlVAE is due to the monotone increasing KL divergence despite having competitively low reconstruction loss in comparison to Big VAE. Conversely, the Big GCVAE‡ (Mahalanobis metric) faces challenges in capturing the correlation between various failure decisions, resulting in a fuzzy latent representation, primarily caused by collapse of the precision matrix. This issue arises due to the high likelihood of similarity in the embedding space, increasing the likelihood of the inverse covariance matrix collapsing. To address this issue, we employ an alternative approach by using the inverse of the diagonal elements rather than the entire covariance inversion.

5.6 Qualitative Evaluation: Big GCVAE

We conduct an evaluation of the generative capabilities of the Big GCVAE models and its variants. The results reveal a notable enhancement in the distribution of BLEU-1, BLEU-3, and LESE-1, LESE-3 scores, as depicted in Figure (7). The figure clearly demonstrates an increased frequency of accurately generated FATs by the model that closely align with the expert failure analysis. This observation is particularly evident in the right-hand side of the same Figure 7, first row. When compared to the decoder-only transformer model (GPT-2), the Big GCVAE exhibits the potential to generate failure analysis sequences that are notably more realistic (following the order of Step type; Substep technique and Equipment). This improvement can be attributed to the Big GCVAE's ability to generalize effectively within the latent embedding space associated with the task.

Subsequently, we conduct a performance comparison between Big GCVAE‡ and GPT2 to assess their generative capabilities and determine whether the generated outputs are plausible decisions that a failure analyst engineer would make. Note that all assessment of the generative strength of each model based on the output for specific failure description is validated by industrial expert. In the next Section 5.6.1, we provide a comprehensive analysis of the failure root cause analysis generated for a given failure description from the expert's perspective.

5.6.1 FATG: Comparing Big GCVAE and GPT-2

- **Grenoble Kostal leakage on GH2**: The failure analysis triplets generated by Big GCVAE for root cause analysis on **product packages** are generally plausible, except for the third triplet (Physical analysis; Optical inspection; Stereomicroscope szx16 (ecn 6590)). The discrepancy in this triplet arises from the fact that the Substep technique, Optical inspection, and Equipment, Stereomicroscope szx16 (ecn 6590), are not appropriate for a Physical analysis (Step type). In other words, the Substep technique and Equipment are correct for the step type Electrical Failure Verification, rather than Physical analysis. On the other hand, the FATs generated by GPT2 are also plausible, but they contain a higher number of redundant failure analysis triplets. Furthermore, the FATs generated by GPT-2 are more focused on root cause analysis within the **silicon**.

- **Grenoble B601 Face ID H9A HCMOS9A WLCSP CSP H9A HCMOS9A CSP gate oxide breakdown Please to find root cause on Face ID issue**: The failure analysis triplets generated by both Big GCVAE and GPT2 are plausible for the respective contexts of product packages and silicon. A surprising ques-

Model	BLEU-1	BLEU-3	MET.	ROUGE-1			ROUGE-L			LESE-1			Lev-1	LESE-3			Lev-3	PPL
				Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1		Prec.	Rec.	F1		
mini-GPT	11.54	7.51	**34.61**	11.22	16.12	12.63	10.19	14.79	11.52	8.11	8.57	7.11	46.69	0.38	0.27	0.30	16.0	-
GPT2-M	21.26	15.47	26.74	30.37	33.28	29.15	27.65	30.4	26.56	21.08	22.06	19.41	43.0	9.23	9.79	8.55	15.0	1.52
Big VAE	21.87	16.15	26.64	35.83	32.57	31.23	**33.05**	29.94	28.73	**25.13**	22.58	21.46	39.0	**11.29**	10.05	9.57	14.0	6.49
Big ControlVAE	22.25	16.38	27.10	**35.96**	33.09	31.55	33.03	30.30	28.89	25.09	22.82	21.54	40.0	11.11	10.21	9.58	14.0	6.98
Big GCVAE†	22.09	16.25	27.01	35.29	32.87	31.17	32.48	30.13	28.58	24.54	22.84	21.36	40.0	10.93	10.19	9.50	15.0	6.94
Big GCVAE‡	**22.53**	**17.00**	**27.63**	35.67	**33.61**	**31.71**	32.79	**30.83**	**29.08**	24.97	**23.45**	**21.79**	39.0	11.18	**10.39**	**9.70**	**14.0**	6.64

Table 3: Model comparison on BLEU (Papineni et al., 2002), ROUGE-1 & L (Lin, 2004), METEOR (MET.) (Banerjee and Lavie, 2005) and LESE (Ezukwoke et al., 2022b). Higher values (in **bold-blue**) is preferred for all metric except **Lev**-n (average n-gram Levenshtein distance). Big GCVAE‡ performs better across all evaluation metric. Observe the approximately 3-point increase in performance of the generative strength for ROUGE-1 and LESE-1 and a comparable increase for the triplet evaluations.

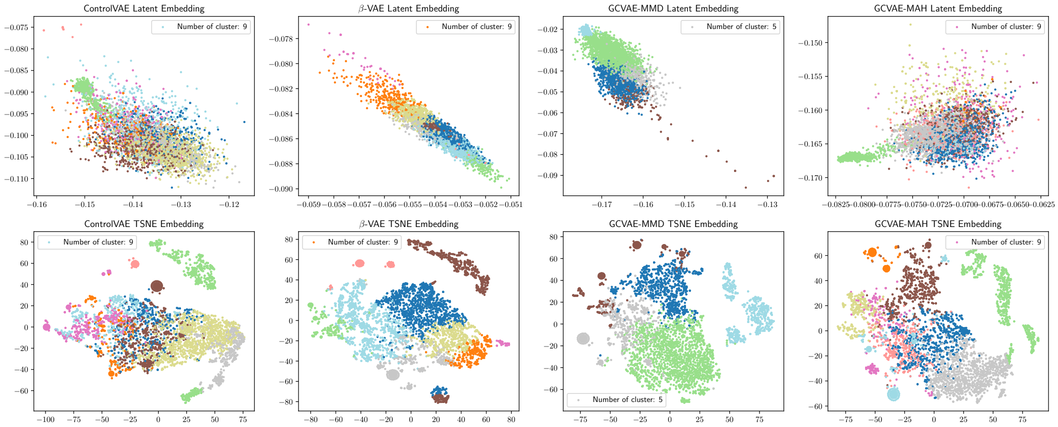
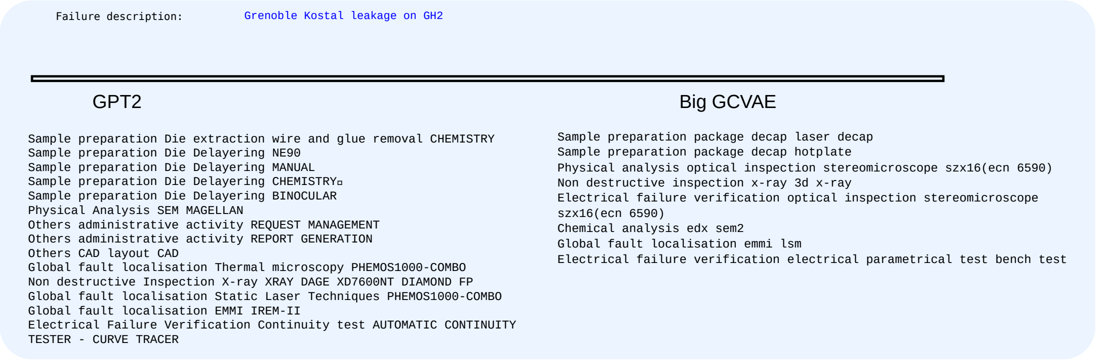


Figure 6: 2D Latent representation space (**top**) and t-SNE Embedding (**bottom**). Observe the quality of clusters in the latent space for Big GCVAE† (best), Big VAE (second best) and Big GCVAE‡ (less fuzzy). The latent space of Big ControlVAE is the most fuzzy with overlapping cluster of densities in t-SNE Embedding space.



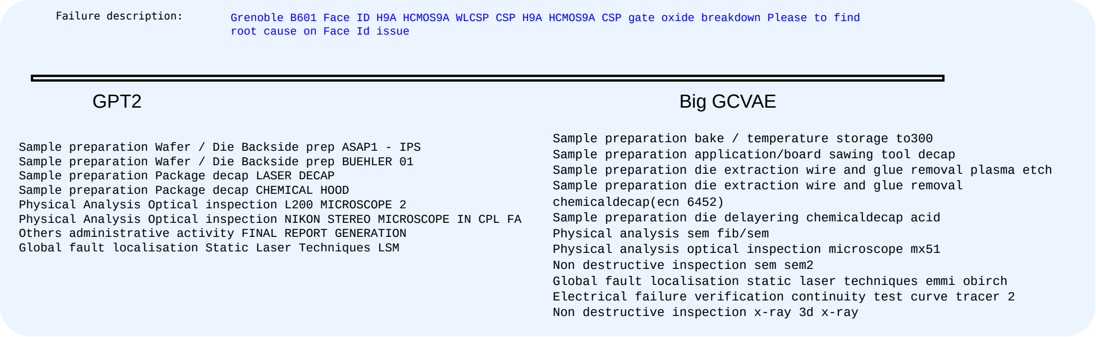
Failure description: Grenoble Kostal leakage on GH2

GPT2

```
Sample preparation Die extraction wire and glue removal CHEMISTRY
Sample preparation Die Delayering NE90
Sample preparation Die Delayering MANUAL
Sample preparation Die Delayering CHEMISTRY□
Sample preparation Die Delayering BINOCULAR
Physical Analysis SEM MAGELLAN
Others administrative activity REQUEST MANAGEMENT
Others administrative activity REPORT GENERATION
Others CAD layout CAD
Global fault localisation Thermal microscopy PHEMOS1000-COMBO
Non destructive Inspection X-ray XRAY DAGE XD7600NT DIAMOND FP
Global fault localisation Static Laser Techniques PHEMOS1000-COMBO
Global fault localisation EMMI IREM-II
Electrical Failure Verification Continuity test AUTOMATIC CONTINUITY
TESTER - CURVE TRACER
```

Big GCVAE

```
Sample preparation package decap laser decap
Sample preparation package decap hotplate
Physical analysis optical inspection stereomicroscope szx16(ecn 6590)
Non destructive inspection x-ray 3d x-ray
Electrical failure verification optical inspection stereomicroscope
szx16(ecn 6590)
Chemical analysis edx sem2
Global fault localisation emmi lsm
Electrical failure verification electrical parametrical test bench test
```



Failure description: Grenoble B601 Face ID H9A HCMOS9A WLCSP CSP H9A HCMOS9A CSP gate oxide breakdown Please to find root cause on Face Id issue

GPT2

```
Sample preparation Wafer / Die Backside prep ASAP1 - IPS
Sample preparation Wafer / Die Backside prep BUEHLER 01
Sample preparation Package decap LASER DECAP
Sample preparation Package decap CHEMICAL HOOD
Physical Analysis Optical inspection L200 MICROSCOPE 2
Physical Analysis Optical inspection NIKON STEREO MICROSCOPE IN CPL FA
Others administrative activity FINAL REPORT GENERATION
Global fault localisation Static Laser Techniques LSM
```

Big GCVAE

```
Sample preparation bake / temperature storage to300
Sample preparation application/board sawing tool decap
Sample preparation die extraction wire and glue removal plasma etch
Sample preparation die extraction wire and glue removal
chemicaldecap(ecn 6452)
Sample preparation die delayering chemicaldecap acid
Physical analysis sem fib/sem
Physical analysis optical inspection microscope mx51
Non destructive inspection sem sem2
Global fault localisation static laser techniques emmi obirch
Electrical failure verification continuity test curve tracer 2
Non destructive inspection x-ray 3d x-ray
```

tion by failure analyst expert is how both models are able to specifically find FATs specific to different context (**product package** and **silicon**). We hypothesize that Big GCVAE

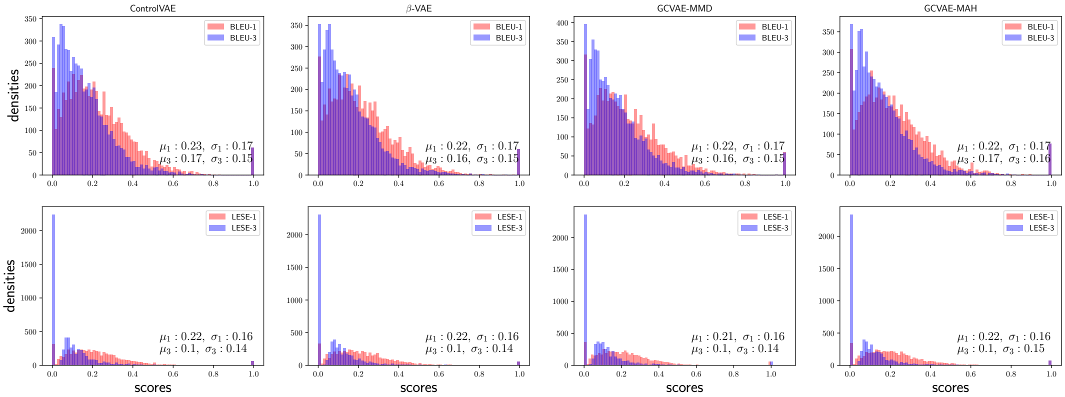


Figure 7: BLUE-1 and BLUE-3 scores distribution (**top**) and LESE-1 and LESE-3 scores distribution (**bottom**) for Big ControlVAE, VAE, GCVAE† and GCVAE‡ models.

generates FATs from the abundance of space it generalizes. This may be a reason why it generates two incorrect sample preparation triplets, including: Sample preparation bake / temperature storage to300 and Sample preparation application/board sawing tool decap.

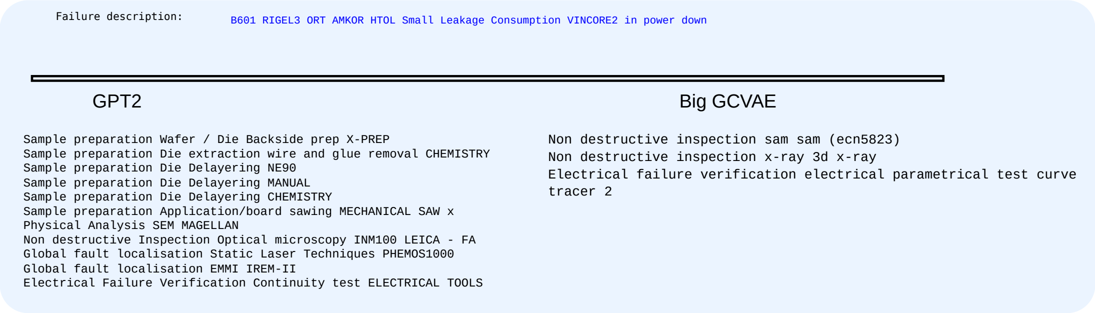
- **B601 RIGEL3 ORT AMKOR HTOL Small Leakage Consumption VINCORE2 in power down**: The failure analysis triplets generated by both Big GCVAE and GPT-2 are entirely incorrect and implausible. This can be attributed to the lack of guiding keywords (or prompts), which are crucial for the models to make accurate predictions for the next words (triplets) in the sequence. The effectiveness of generative models relies on the ability to associate seed words with relevant data points in the embedding space in order to generate meaningful and contextually appropriate tokens.

- **PPM ASSESSMENT F9V 4MEG CROSS-SECTION**: Similar to the previous FATG challenge, the failure analysis triplets generated in this case are incorrect and lack plausibility for the given failure description. This is primarily due to the absence of the keyword **cross-section** in the proposed sets of analysis, and since no cross-section keyword is found in the generated FATs, it is a hallucinated decision. Including this keyword with sufficient context in the failure description is essential for generating relevant and accurate failure analysis triplets that align with the specified failure scenario.

This paper presents real-world use cases to demonstrate the generative efficacy of Big GCVAE, a large-scale language model that has been fine-tuned on failure analysis data from the semiconductor industry using an adaptive loss function called GCVAE. Our findings highlight that although the model may not consistently generate precise failure analysis sequences specific to a given failure description, it showcases a robust representation space that can be further enhanced by providing high-quality failure description keywords and an initial failure analysis triplet as a seed. This underscores the potential of Big GCVAE in assisting failure analysis engineers during the decision-making process for failure root cause analysis.

In our comparative analysis between Big GCVAE and GPT2, we observed that Big GCVAE generally demonstrates superior generative performance. The model exhibits the ability to generate more coherent and contextually appropriate failure analysis triplets for decision-making during failure root cause analysis.

However, despite its overall proficiency, Big GCVAE faces occasional challenges in accurately capturing the correct sequence of triplets from the latent space z. As a result, it may produce unrealistic triplets that do not align with the expected context or domain-specific knowledge. An example of such a limitation is when the model erroneously associates a Substep technique, such as **Optical inspection**, with an inappropriate Step type, such as Physical Analysis, instead of the correct Electrical Failure Verification type. These occurrences indicate that while Big GCVAE excels in generating meaningful triplets, there is still room for further enhancement in its ability to ensure precise corre-



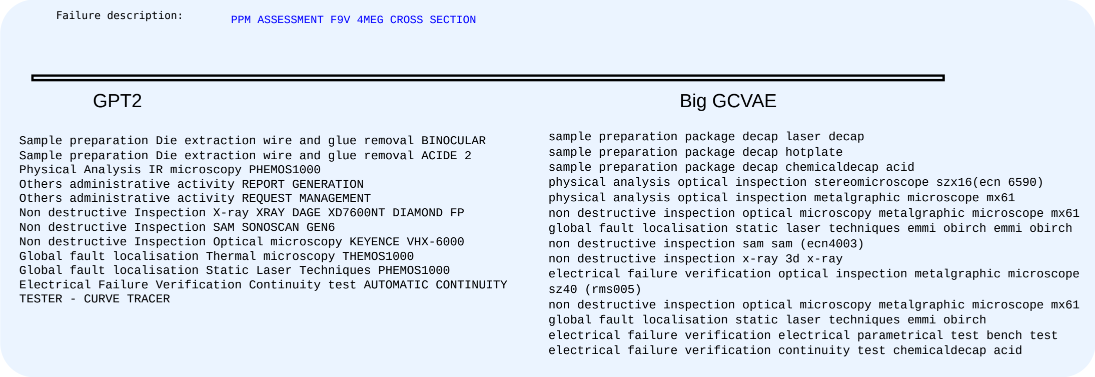
Failure description: B601 RIGEL3 ORT AMKOR HTOL Small Leakage Consumption VINCORE2 in power down

GPT2

```
Sample preparation Wafer / Die Backside prep X-PREP
Sample preparation Die extraction wire and glue removal CHEMISTRY
Sample preparation Die Delayering NE90
Sample preparation Die Delayering MANUAL
Sample preparation Die Delayering CHEMISTRY
Sample preparation Application/board sawing MECHANICAL SAW x
Physical Analysis SEM MAGELLAN
Non destructive Inspection Optical microscopy INM100 LEICA - FA
Global fault localisation Static Laser Techniques PHEMOS1000
Global fault localisation EMMI IREM-II
Electrical Failure Verification Continuity test ELECTRICAL TOOLS
```

Big GCVAE

```
Non destructive inspection sam sam (ecn5823)
Non destructive inspection x-ray 3d x-ray
Electrical failure verification electrical parametrical test curve
tracer 2
```



Failure description: PPM ASSESSMENT F9V 4MEG CROSS SECTION

GPT2

```
Sample preparation Die extraction wire and glue removal BINOCULAR
Sample preparation Die extraction wire and glue removal ACIDE 2
Physical Analysis IR microscopy PHEMOS1000
Others administrative activity REPORT GENERATION
Others administrative activity REQUEST MANAGEMENT
Non destructive Inspection X-ray XRAY DAGE XD7600NT DIAMOND FP
Non destructive Inspection SAM SONOSCAN GEN6
Non destructive Inspection Optical microscopy KEYENCE VHX-6000
Global fault localisation Thermal microscopy THEMOS1000
Global fault localisation Static Laser Techniques PHEMOS1000
Electrical Failure Verification Continuity test AUTOMATIC CONTINUITY
TESTER - CURVE TRACER
```

Big GCVAE

```
sample preparation package decap laser decap
sample preparation package decap hotplate
sample preparation package decap chemicaldecap acid
physical analysis optical inspection stereomicroscope szx16(ecn 6590)
physical analysis optical inspection metalgraphic microscope mx61
non destructive inspection optical microscopy metalgraphic microscope mx61
global fault localisation static laser techniques emmi obirch emmi obirch
non destructive inspection sam sam (ecn4003)
non destructive inspection x-ray 3d x-ray
electrical failure verification optical inspection metalgraphic microscope
sz40 (rms005)
non destructive inspection optical microscopy metalgraphic microscope mx61
global fault localisation static laser techniques emmi obirch
electrical failure verification electrical parametrical test bench test
electrical failure verification continuity test chemicaldecap acid
```

spondence between the generated triplets and their relevant context within the failure analysis domain.

6 Conclusion

To overcome the challenges of robust representation and high-quality generation of failure analysis triplets, we propose a new approach that involves fine-tuning a Transformer-based Variational Autoencoder (VAE) architecture using an unmasked pre-trained BERT Encoder and a GPT2 Decoder. By leveraging the Generalized-Controllable Variational AutoEncoding (GCVAE) loss, our model aims to achieve an optimized representation with a low reconstruction loss and highly disentangled latent space. Our evaluation of the model's performance in generating failure analysis triplets yields the following key findings:

- Big GCVAE can generate failure analysis triplets that are logical and reasonable, providing valuable insights for expert failure analyst engineers in the decision-making process.
- The model demonstrates its ability to generate failure analysis triplets specifically tailored to root cause identification in product packages, and it can also address potential root causes within a Silicon. This is made possible by the model's ability to generalize and draw predictions from the embedding latent space.
- Notably, Big GCVAE is a self-supervised model that operates with adaptive-controllable hyperparameters, eliminating the need for human intervention in the decision-making process.

In summary, Big GCVAE is a robust model that can generate failure analysis triplets (sequences of text-encoded steps for analyzing defective components in the semiconductor industry) that are logical, reasonable, and tailored to specific problems. The model is able to do this by learning to represent failure analysis triplets in a latent space that is both disentangled and informative. Additionally, Big GCVAE is a self-supervised model, meaning that it can be trained without the need for human-labeled data.

Big GCVAE has the potential to be a valuable tool for failure analyst engineers in the semiconductor industry. By providing them with logical and reasonable failure analysis triplets, Big GCVAE can help them to identify root causes more quickly and accurately. Additionally, Big GCVAE's ability to generalize and draw predictions from the embedding latent space makes it a powerful tool for addressing new and emerging failure scenarios.

7 Limitation

Despite the overwhelming performance of Big GC-VAE (BERT-GPT2) model for the task of failure analysis triplets generation, it stills suffers significant challenge that can be addressed. It is crucial to acknowledge that the model may occasionally generate unrealistic failure analysis triplets due to the phenomenon of hallucination. This can be both a problem of overgeneralization and overfitting. However, no particular metric perfectly addresses this phenomenon, except the quantitative and domain expert evaluations mentioned in sections 5.5 and 5.6 respectively. This limitation highlights the need for further refinement and improvement by prompt engineering failure description and using reinforcement learning to mitigate the occurrence of unrealistic outputs.

Acknowledgments

The authors would like to thank the industrial collaborators and failure analyst experts from STMicroelectronics, for their useful information on the data, discussion during live demonstration and validation of results. Special thanks to the project sponsor, EURIPIDES[2] under the PENTA framework for the Failure Analysis (FA4.0) EUREKA Cluster.

Competing Interest

There are Non-financial competing interest including: STMicroelectronics (st.com), Infineon (infineon.com), Robert Bosch GmbH (bosch.com), Ericsson (ericsson.com), Université Jean Monnet (univ-st-etienne.fr), Mines Saint-Étienne (emse.fr), Orsay Physics (orsayphysics.com), TESCAN ORSAY HOLDING (tescan-orsay.com), University of Stuttgart (uni-stuttgart.de), Fraunhofer Institute for Microstructure of Material and Systems IMWS (fraunhofer.de), Materiex AB (materiex.com) and Matworks GmbH (matworks.de).

Funding

Funding is provided for the Failure Analysis (FA4.0) project by two EUREKA Clusters, EURIPIDES[2] and PENTA with project profile number 19007.

Author Contribution

The primary contributor to this project is Kenneth Ezukwoke, who took the lead in conducting both the research and experimentation. The study is under the supervision of Mireille Batton-Hubert, Anis Hoayek, and Xavier Boucher, who have contributed by reviewing the paper and providing insights on translating the experimentation results to address questions related to industrial use cases. Pascal Gounet and Jérôme Adrian, Failure Analyst Engineers from STMicroelectronics, played a crucial role in validating the quantitative results and assessing their practical significance.

Data Availability Statement

The experiment described in this paper utilizes a confidential (proprietary) data source obtained from STMicroelectronics. The data is exclusive to the collaborating microelectronics manufacturers involved in the FA4.0 project. As the data source is confidential, specific details regarding its composition are not disclosed in this paper. However, the variables utilized in the experiment are listed and described in the Dataset section 5.3 of this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Stefan Biffl and Michael Halling. 2003. Investigating the defect detection effectiveness and cost benefit of nominal inspection teams. *IEEE Transactions on Software Engineering*, 29(5):385 – 397. Cited by: 64.

Manal Binkhonain and Liping Zhao. 2019. A review of machine learning algorithms for identification and classification of non-functional requirements. *Expert Systems with Applications: X*, 1:100001.

Baidyanath Biswas, Pooja Sengupta, Ajay Kumar, Dursun Delen, and Shivam Gupta. 2022. A critical assessment of consumer reviews: A hybrid nlp-based methodology. *Decision Support Systems*, 159:113799.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. 2016. Importance weighted autoencoders. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

T. Caliński and J Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27.

Yu Chen and Mohammed J. Zaki. 2017. Kate: K-competitive autoencoder for text. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

David L. Davies and Donald W. Bouldin. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Kenneth Ezukwoke, Anis Hoayek, Mireille Batton-Hubert, and Xavier Boucher. 2022a. Gcvae: Generalized-controllable variational autoencoder.

Kenneth Ezukwoke, Anis Hoayek, Mireille Batton-Hubert, Xavier Boucher, Pascal Gounet, and Jerome Adrian. 2022b. Leveraging pre-trained models for failure analysis triplets generation.

Kenneth Ezukwoke, Houari Toubakh, Anis Hoayek, Mireille Batton-Hubert, Xavier Boucher, and Pascal Gounet. 2021. Intelligent fault analysis decision flow in semiconductor industry 4.0 using natural language processing with deep clustering. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 429–436.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019a. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 240–250, Minneapolis, Minnesota. Association for Computational Linguistics.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019b. Cyclical annealing schedule: A simple approach to mitigating kl vanishing.

Rachit Garg, Arvind W Kiwelekar, Laxman D Netak, and Akshay Ghodake. 2021. i-pulse: A nlp based novel approach for employee engagement in logistics organization. *International Journal of Information Management Data Insights*, 1(1):100011.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.*, 61(1):65–170.

I. Higgins, L. Matthey, A. Pal, Christopher P. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, ACL '83, page 145–150, USA. Association for Computational Linguistics.

Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. 2019. A surprisingly effective fix for deep latent variable modeling of text.

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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, page 71–78, USA. Association for Computational Linguistics.

D. Liu and G. Liu. 2019. A transformer-based variational autoencoder for sentence generation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

Genggeng Liu, Canyang Guo, Lin Xie, Wenxi Liu, Naixue Xiong, and Guolong Chen. 2020. An intelligent cnn-vae text representation technology based on text semantics for comprehensive big data.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2873–2879. AAAI Press.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences.

Grzegorz Loniewski, Emilio Insfran, and Silvia Abrahão. 2010. A systematic review of the use of requirements engineering techniques in model-driven development. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6395 LNCS(PART 2):213 – 227. Cited by: 79.

Villacourt Mario. 1992. Failure reporting, analysis and corrective action system in the us semiconductor manufacturing equipment industry: A continuous improvement process. In *Thirteenth IEEE/CHMT International Electronics Manufacturing Technology Symposium*, pages 111–115.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6297–6308, Red Hook, NY, USA. Curran Associates Inc.

Barak Oshri. 2015. There and back again : Autoencoders for textual reconstruction.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Mehrdokht Pournader, Hadi Ghaderi, Amir Hassanzadegan, and Behnam Fahimnia. 2021. Artificial intelligence applications in supply chain management. *International Journal of Production Economics*, 241:108250.

Quoc V. Le Prajit Ramachandran, Peter J. Liu. 2016. Unsupervised pretraining for sequence to sequence learning. *arXiv*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pretraining.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Abbas Rammal, Kenneth Ezukwoke, Anis Hoayek, and Mireille Batton-Hubert. 2023a. Root cause prediction for failures in semiconductor industry, a genetic algorithm-machine learning approach. *Scientific Reports*, 13:4934.

Abbas Rammal, Kenneth Ezukwoke, Anis Hoayek, and Mireille Batton-Hubert. 2023b. Unsupervised approach for an optimal representation of the latent space of a failure analysis dataset. *The Journal of Supercomputing*, pages 1–27.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2).

Huajie Shao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Dongxin Liu, Jun Wang, and Tarek Abdelzaher. 2020. Controlvae: Controllable variational autoencoder.

Reza Toorajipour, Vahid Sohrabpour, Ali Nazarpour, Pejvak Oghazi, and Maria Fischl. 2021. Artificial intelligence in supply chain management: A systematic literature review. *Journal of Business Research*, 122:502–517.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhiqiang Wang, Kenneth Ezukwoke, Anis Hoayek, Mireille Batton-Hubert, and Xavier Boucher. 2022. Nlp based on gcvae for intelligent fault analysis in semiconductor industry. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, page 1–8. IEEE Press.

Yonghui Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason R. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Yijun Xiao, Tiancheng Zhao, and William Yang Wang. 2018. Dirichlet variational autoencoder for text modeling.

G. Yue, G. Ping, and L. Lanxin. 2018. An end-to-end model based on cnn-lstm for industrial fault diagnosis and prognosis. In *2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, pages 274–278.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders.

In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada. Association for Computational Linguistics.

M. Zimmer, A. Al-Yacoub, P. Ferreira, and N. Lohse. 2019. Understanding human decision-making during production ramp-up using natural language processing. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 337–342.