machine learning & knowledge extraction

MDPI

Article

Benefits from Variational Regularization in Language Models

Cornelia Ferner * and Stefan Wegenkittl

Information Technology and Systems Management, Salzburg University of Applied Sciences, Urstein Sued 1, 5412 Puch/Hallein, Austria; stefan.wegenkittl@fh-salzburg.ac.at

* Correspondence: cornelia.ferner@fh-salzburg.ac.at

Abstract: Representations from common pre-trained language models have been shown to suffer from the degeneration problem, i.e., they occupy a narrow cone in latent space. This problem can be addressed by enforcing isotropy in latent space. In analogy with variational autoencoders, we suggest applying a token-level variational loss to a Transformer architecture and optimizing the standard deviation of the prior distribution in the loss function as the model parameter to increase isotropy. The resulting latent space is complete and interpretable: any given point is a valid embedding and can be decoded into text again. This allows for text manipulations such as paraphrase generation directly in latent space. Surprisingly, features extracted at the sentence level also show competitive results on benchmark classification tasks.

Keywords: language models; regularization; isotropy; generalizability; semantic reasoning

Citation: Ferner, C.; Wegenkittl, S. Benefits from Variational Regularization in Language Models. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 542–555. https://doi.org/10.3390/make4020025

Academic Editors: Irena Spasić and Andreas Holzinger

Received: 29 April 2022
Accepted: 8 June 2022
Published: 9 June 2022

1. Introduction

Self-supervised, attention-based language models are prone to overfitting or memorizing input data, especially when trained on smaller datasets. Nevertheless, models such as BERT [1], based on an encoder-only architecture and trained on very large datasets, exhibit state-of-the-art performance on common natural language processing tasks. The resulting token representations of these models are distributed and highly contextual, but the latent space does not exhibit additional structural properties such as isotropy and the resulting smoothness and completeness [2].

Previous work has shown that isotropic latent representations, which are distributed across latent space instead of populating a narrow cone [2], can improve the performance of language models on transfer tasks. Gao et al. [3] improved the performance of a Transformer-based model with increased isotropy by directly optimizing the cosine distance between the latent representations. Wang et al. [4] suggested controlling the singular-value distribution of the output representation matrix, and Li et al. [5] transformed the original representation distribution into a Gaussian distribution through normalizing flows to increase the isotropy of the underlying models. In addition to a higher isotropy, we aim for a so-called complete latent space, where not only vectors are close to training examples, but any arbitrary latent representation can be transformed back to a meaningful original representation. This calls for an autoencoder architecture with a decoder network that is able to decode all latent representations to sentences again.

In order to enforce isotropic and complete latent representations, we apply a regularizing constraint during training, similar to Gao et al. [3]. A common regularization network is the so-called Variational Autoencoder (VAE) [6]. VAE networks consist of an encoder that maps given input data not to a point in latent space but a distribution. A VAE's decoder is required to successfully reconstruct the original input from samples of the latent distribution. Thus, the VAE is optimized to reconstruct the given input sequence whilst enforcing the latent distributions to match a given prior distribution.

We propose a Variational Auto-Transformer (VAT), a VAE based on a Transformer architecture, where the variational loss is computed at the token level. We show that by enforcing a Gaussian distribution as the latent prior, the latent token-level representations

become more isotropic in comparison to models such as BERT. We introduce the prior distribution's standard deviation as a model parameter to optimize isotropy and balance the language generation variety against the network's reconstruction ability.

In order to be able to demonstrate the completeness of the latent space, our model differs from BERT-like models in that it contains a decoder network. The pre-trained decoder can map back every point in latent space to a textual representation, be it actually encoded or synthetic, sampled latent points. This allows for text generation through "variational sampling" and other manipulations such as interpolation directly in latent space. While the primary goals of our model are tasks that require paraphrase generation, the resulting encoders surprisingly also perform on par in transfer tasks with smaller training setups. We also show that sentence-level representations, e.g., obtained through averaging, are suitable for sentence classification tasks.

2. Background

An autoencoder is a neural network architecture consisting of an encoder *enc*, which maps any input $\boldsymbol{x} \in \mathbb{R}^d$ to a point $\boldsymbol{z} = enc(\boldsymbol{x})$ in latent space, and a decoder *dec*, such that $\boldsymbol{x} \approx dec(enc(\boldsymbol{x}))$ [7]. The chosen network architecture defines the families E and D of the encoder and decoder. During the training of the autoencoder, the network parameters are optimized to find the optimal (enc^*, dec^*) pair that minimizes a given loss function $\mathcal{L}$:

$$(enc^*, dec^*) = \underset{(enc,dec)\in E\times D}{\arg\min} \; (\mathcal{L}) \tag{1}$$

For the standard autoencoder with continuous inputs $\boldsymbol{x} \in \mathbb{R}^d$, the reconstruction loss $\mathcal{L} = \mathcal{L}_{REC}$ is

$$\mathcal{L}_{REC} = \|\boldsymbol{x} - dec(enc(\boldsymbol{x}))\|_2 \tag{2}$$

Over-complete autoencoder architectures with many degrees of freedom are prone to a particular form of overfitting: the encoder maps each data point $\boldsymbol{x}$ to an isolated point $\boldsymbol{z}$ in the latent space such that the decoder memorizes the lossless reconstruction of each of these codes. This highly discrete latent space lacks completeness and is of limited use for advanced NLP applications.

Variational Autoencoder

The Variational Autoencoder (VAE) [6] can be thought of as a generative autoencoder. A VAE encodes an input data point as a distribution over the latent space by adding a regularization term to the loss function. The regularization term $\mathcal{L}_{REG}$ assesses the Kullback–Leibler divergence between the latent distribution and a standard Gaussian based on their means $\boldsymbol{\mu}$ and covariances Σ:

$$\mathcal{L}_{REG} = D_{KL}[N(\boldsymbol{\mu}, \Sigma), N(\mathbf{0}, I)] \tag{3}$$

$\boldsymbol{\mu}$ and Σ are either estimated using standard point estimators from the encoder outputs or computed by learned functions such that $\boldsymbol{\mu} = g(\boldsymbol{x})$ and $\Sigma = h(\boldsymbol{x})$ with $g \in G$ and $h \in H$, where G and H are families of network architectures. G and H can contribute to the desired properties of the latent representation by implicitly implementing dimensionality reduction or feature disentangling.

Decoding from the latent representation requires sampling from $\boldsymbol{z} \sim N(\boldsymbol{\mu}, \Sigma)$. In order to enable backpropagation despite this sampling operation, Kingma and Welling [6] introduced the reparameterization trick: instead of sampling from the latent distribution, a random sample $\boldsymbol{\epsilon} \sim N(\mathbf{0}, I)$ from a standard Gaussian is drawn and then transformed by the computed mean and standard deviation $\boldsymbol{z} = L^T\boldsymbol{\epsilon} + \boldsymbol{\mu}$, where L^T is the Cholesky factor of Σ.

The VAE's total loss is the sum of the regularization and reconstruction term:

$$(enc^*, dec^*) = \underset{(enc,dec)\in E\times D}{\arg\min} \left(\mathcal{L}_{REC} + \mathcal{L}_{REG}\right) \tag{4}$$
$$= \underset{(enc,dec)\in E\times D}{\arg\min} \left(\|\boldsymbol{x} - dec(\boldsymbol{z})\|_2 + D_{KL}[N(\boldsymbol{\mu}, \Sigma), N(\mathbf{0}, I)]\right)$$

3. Related Work

The architecture of variational autoencoders with different kinds of networks has previously been optimized for various natural language processing tasks. The models discussed below focus on optimizing either for semantic similarity and text classification tasks or for the generation of text. Some of the models no longer contain a decoder, such that the latent representations are not interpretable. The variational loss is applied to a sentence-level latent representation in all these models. Reasoning about token contextuality or sampling of individual tokens thus is not possible.

Several models were suggested to solve common document or text classification tasks: Gururangan et al. [8] used MLPs as the encoder and decoder to learn latent vectors from bag-of-words inputs that were used for document classification. Mahabadi et al. [9] compressed BERT-based sentence representations through a Gaussian prior distribution for transfer classification tasks. Deudon [10] implemented a Siamese network architecture based on sentence representations from a VAE with Bi-LSTMs for semantic similarity classification.

Zhao et al. [11] used an LSTM-based VAE for dialog generation. Miao et al. [12] relied on MLPs in a VAE for both document modeling and answer selection. Wang et al. [13] introduced a Transformer-based VAE for story completion, where the missing plot is conditioned on the latent representation capturing the distribution of the story plot. Shu et al. [14] proposed a Transformer-based non-autoregressive model specifically for machine translation that incorporates a predictor network for the length of the target sentence.

Various network architectures have been proposed for the more general task of language modeling. Bowman et al. [15] applied the variational loss on the last hidden state of an LSTM sequence-to-sequence model. In iVAE [16], an MLP produces sample-based distributions from an LSTM's hidden representation concatenated with random noise. Yang et al. [17] applied an LSTM as the encoder and a CNN as the decoder in a language model. Liu and Liu [18] added a feed-forward layer to a Transformer model to map the encoder outputs to a mean and variance vector, which were then upsampled and passed to the decoder followed by an LSTM layer. OPTIMUS [19] extends this idea to large-scale language models. The encoder weights are initialized with pre-trained BERT weights, the decoder with those from a pre-trained GPT-2 model. The latent representation of the start token is used as the sentence-level representation.

Token-level regularization has been previously applied to RNNs for language modeling [20]. Similarly, the variational loss at the token level for the Variational Auto-Transformer allows for direct manipulations in the latent space, resulting in a range of possibilities for text generation. Sentence-level representations can still be obtained and are suitable for sentence classification tasks.

4. Variational Auto-Transformer

Initially proposed for machine translation, the Transformer [21] can be used as the language model and framed as an autoencoder. Extended with a token-level variational loss, the VAT maps input tokens to context-aware, distributed latent representations. Figure 1 illustrates the architecture of the VAT. The set of encoders E and decoders D is defined by the Transformer's network architecture based on self-attention modules. The encoder maps the embedded input sequence $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$, $X \in \mathbb{R}^{T\times d}$ to a latent representation $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_T\}$, $Z \in \mathbb{R}^{T\times d}$, where T is the sequence length and d is the model dimension. In order to do so, the model predicts $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}_t$ from $\boldsymbol{x}_t$ and samples $\boldsymbol{z}_t \sim N(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$. Z is

then used as the attention source in the decoder to predict the next word based on the already produced output. As X and Z depend on T, the objective is to minimize

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}_{q(Z|X)}\|\boldsymbol{x}_t - dec(\boldsymbol{z_t})\|_2 + D_{KL}[q(\boldsymbol{z}_t|X), p(\boldsymbol{z}_t)]\right) \tag{5}$$

where $p(\boldsymbol{z}_t) \sim N(\mathbf{0}, I)$ and $q(\boldsymbol{z}_t|X) \sim N(\boldsymbol{\mu}_t, \Sigma_t)$. More precisely, we let $\Sigma_t = \mathrm{diag}(\exp \boldsymbol{\sigma}_t)$, where the vector $\boldsymbol{\sigma}_t = (\sigma_j)_{j=1\ldots d}$ is predicted, so that computing the actual value of Σ involves the exponential as a nonlinear activation function.

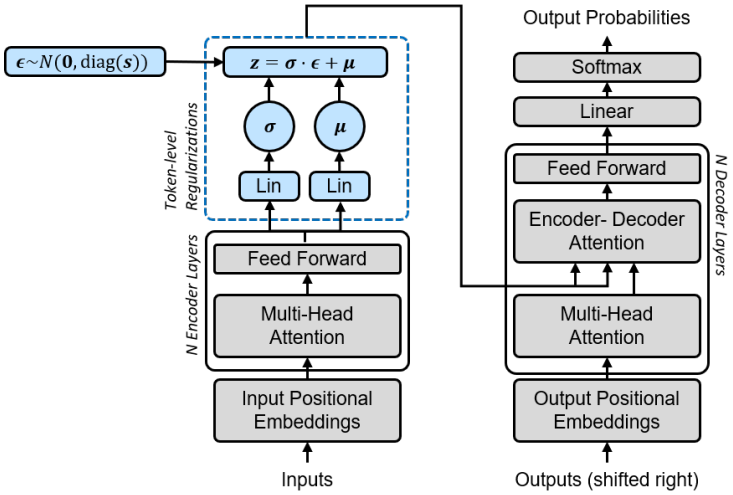
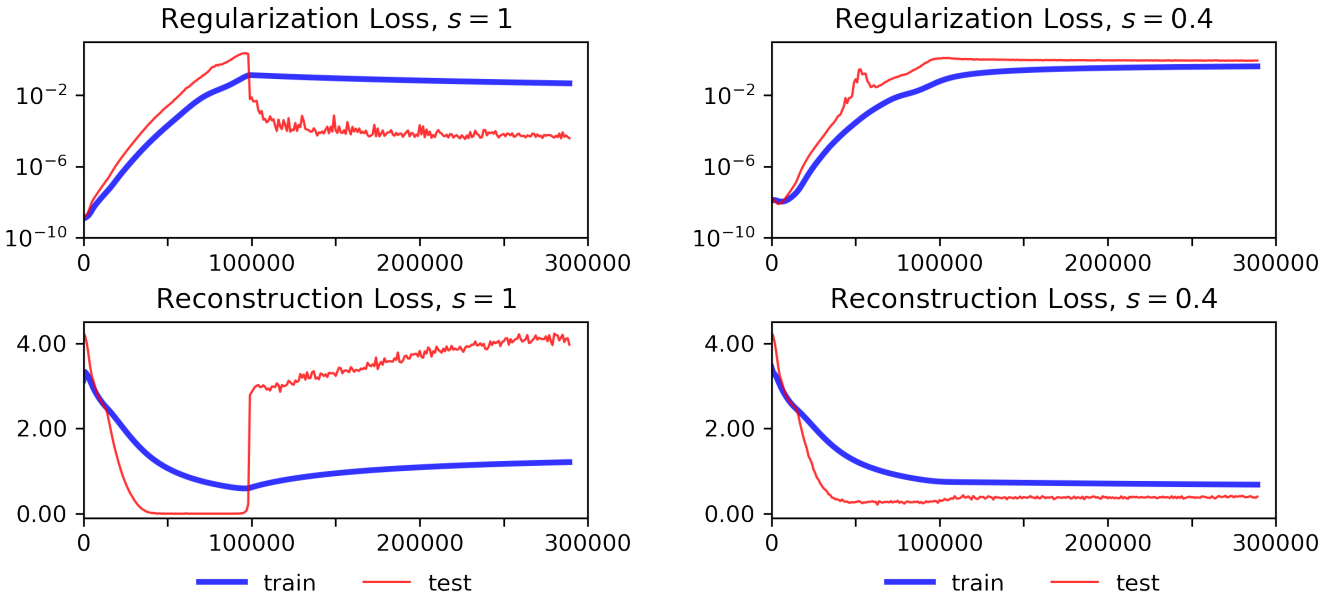


Figure 1. The proposed Variational Auto-Transformer architecture: Mean and covariance for the variational loss (blue, dashed lines) are computed through two independent linear layers. The encoder and decoder (grey, solid lines) architecture is the same as in the original Transformer.

The output of the encoder is passed to two linear layers in parallel. The linear layers have the same number of input and output nodes and learn a transformation to predict $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}_t$ for each token. The latent representation Z is used as the input in the encoder–decoder attention layer of the Transformer's decoder.

Scaling the Regularization Loss

During our experiments, we experienced overfitting on the training data for the reconstruction loss. The learning curves of a corresponding experiment are illustrated in Figure 2a. Weighting the regularization loss according to a logistic annealing function as suggested by Bowman et al. [15] or by a scaling factor β similar to the scaling of the beta-VAE [22], but with $\beta < 1$, only had a small effect.



(**a**) Standard Gaussian $N(\mathbf{0}, \boldsymbol{I})$ (**b**) Gaussian $N(\mathbf{0}, \text{diag}(0.4))$

Figure 2. Weighted regularization (**top**) and reconstruction loss (**bottom**) for training and test data over several epochs for two different Gaussians applied as a prior distribution for the regularization term.

Thus, we propose to scale the covariance matrix of the target distribution of $\mathcal{L}_{REG}$ instead. This is motivated by the observation that standard Gaussians in d-dimensional latent spaces (e.g., $d \geq 128$) will overlap considerably if their mean values are—at the same time—regularized to zero. As a consequence, $\mathcal{L}_{REG}$ cannot be minimized by the model without increasing $\mathcal{L}_{REC}$ in an undesirable way. Scaling the standard deviation to a too small value, though, might result in peaky distributions, having no regulatory effect and resulting in a trivial reconstruction objective.

We adapted the computation of the regularization loss to incorporate a scaling factor s for the standard deviation as the hyperparameter. The closed-form of the VAE loss function (refer to Odaibo [23] for the derivation of the closed form) with a prior standard deviation $\sigma_p = s \in (0, 1]$ and prior $\mu_p = 0$ for a specific token's layer outputs σ_t and $\boldsymbol{\mu}_t$ becomes

$$-D_{KL}[N(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t), N(\mathbf{0}, diag(s))] = \sum_{i=1}^{d} \frac{1}{2} \log(\sigma_{t,i}^2) - \frac{\sigma_{t,i}^2 + \mu_{t,i}^2}{2s^2} - \log(s) + \frac{1}{2} \tag{6}$$

This results in codes $\boldsymbol{z}_t = \boldsymbol{\sigma}_t \cdot \boldsymbol{\epsilon}'_t + \boldsymbol{\mu}_t$ being distributed according to a Gaussian with zero mean and a predefined standard deviation, $\boldsymbol{\epsilon}' \sim N(\mathbf{0}, \text{diag}(\boldsymbol{s}))$. The optimal balance between $\mathcal{L}_{REC}$ and $\mathcal{L}_{REG}$ as expressed by the value of $s \in (0, 1]$ is related to the representations' isotropy and will be determined experimentally.

5. Experiments

The aim of the experiments is to balance the two loss terms. $\mathcal{L}_{REC}$ should be low to maintain the reconstruction ability of the network, while a too small scaling value s implies that the VAT essentially behaves as a regular AE, decoding from $\boldsymbol{z} = \boldsymbol{\mu}$. At the same time, we want to optimize the isotropy of the latent representations. In order to choose the best setting, we observe both the resulting loss values and the properties of the latent space in terms of similarity between representations.

Our VAT model architecture is smaller than the original Transformer architecture, as it is not intended for machine translation, but autoencoding. The VAT model with dimensionality $d = 128$ consists of $N = 4$ encoder and decoder layers, respectively, with

$H = 8$ attention heads each. The dimension in the feed-forward layers is 512. Dropout during training was set to 0.1. The Noam optimizer [21] operates with 50,000 warmup steps. A logistic annealing function [15] with 50,000 warmup steps and an initial value of 0.00025 was applied to weight $\mathcal{L}_{REG}$.

The VAT was trained on the training split of the WMT19 de-en dataset [24] using English sentences only. WMT19 contains data from news commentaries, Wiki titles, Europarl, ParaCrawl, and Common Crawl corpora. The data are tokenized using subword tokenization with a target vocabulary of 2^{15} tokens. The batch size was set to 128. Decoded sentences that are presented as results are obtained through beam search with beam size 5.

Optimal Scaling Parameter

The isotropy of the learned representations z is assessed as a function of the scaling factor $s \in (0, 1]$. Ethayarajh [2] introduced the notion of isotropy of the latent space as the mean cosine similarity between vector representations of random tokens. In an isotropic latent space, representations of randomly sampled tokens have low cosine similarity and do not cluster in a specific direction.

In addition to isotropy, Ethayarajh [2] introduced the notion of self-similarity and intra-sentence similarity. A high self-similarity, i.e., mean cosine similarity between representations of the same token in different sentences, indicates that neighboring representations capture similar concepts, which contributes to the smoothness of the latent space. Intra-sentence similarity measures the mean cosine similarity of tokens occurring in the same sentence to the mean sentence representation, thus being related to contextuality.

BERT exhibits low isotropy (mean cosine similarity > 0.4) in its last layers [2], a phenomenon known as representation degeneration [3]. In contrast to BERT with high contextuality with respect to the low isotropy, we expect a more complete latent space for the VAT and, thus, high isotropy. The results in Figure 3a were computed for the STS12 datasets [25]. Different from Ethayarajh [2], Figure 3a depicts the plain intra-sentence and self-similarity, i.e., not adjusted for isotropy. The average cosine similarity between randomly sampled words is low for smaller s values, suggesting an isotropic latent space with representations distributed across space. The latent space is most isotropic at $s = 0.4$.

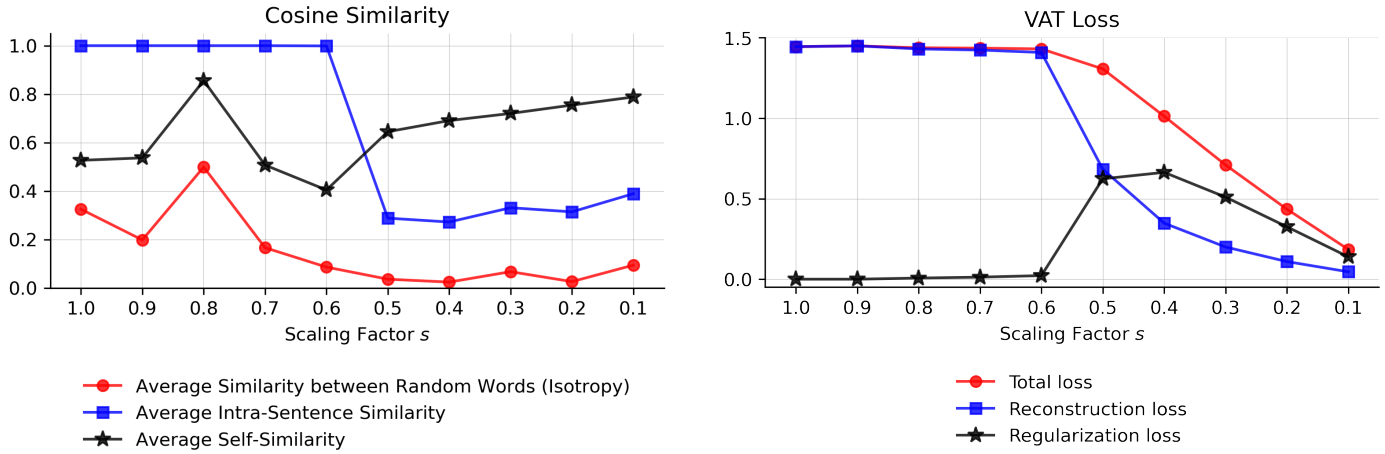


Figure 3. Optimizing the scaling factor s with respect to isotropy and loss values. (**a**) Cosine similarities of the latent representations of the VAT at the end of the training process denoting the isotropy of the latent space and the contextuality of token representations; (**b**) reconstruction loss, weighted regularization loss, and total loss of the VAT at the end of the training process.

The average self-similarity of tokens greater than 0.6 suggests a mapping of similar concepts into the same region, indicating smoothness in the latent space. The intra-sentence similarity for $0.5 \geq s \geq 0.1$ is lower than the self-similarity, but still above the similarity of

random words. This indicates that contextual information is captured in the representations. Interestingly, for $1 \geq s \geq 0.6$, the token representations seem to "collapse" into an identical representation for all positions in a sequence, resulting in an intra-sentence similarity of 1. Among the words with most context-specific representations are mainly stopwords, as they have the lowest self-similarity.

In Figure 3b, we report the final loss values after training the models for 3 epochs with different scaling factors $s \in (0, 1]$. Scaling values $1 \geq s \geq 0.6$ result in near-zero regularization losses and high reconstruction error. The original input cannot be reproduced any more. This observation corresponds to the findings from Figure 3a, where the representations within a sentence are identical. Starting from $s = 0.5$, the reconstruction loss drops, and with $s = 0.4$, the reconstruction loss falls below the regularization loss. Values smaller than $s = 0.4$ further decrease the positive effects of the variational model, as the achievable variety of generated text is reduced. This phenomenon is referred to as posterior collapse [26].

Combining the findings from evaluating the loss and latent space properties in Figure 3, we selected $s = 0.4$ as the scaling factor for further evaluating the VAT. This value corresponds to good reconstruction ability and isotropic representations, which impacts both the generation of text, as well as the performance on classification tasks. Furthermore, the learning curves corresponding to $s = 0.4$ no longer exhibit overfitting (see Figure 2b).

6. Variational Language Model

During training, the VAT encoder produces variants of the input tokens by drawing from the latent distribution. The VAT decoder has to be able to reproduce the original sequence given variational input. At test time, both approaches are possible: Decoding with the standard deviation set to 0, i.e., $\boldsymbol{z}_t = \boldsymbol{\mu}_t$, leads to a deterministic representation and is used for classification tasks. For sequence generation, decoding $\boldsymbol{z}_t = \boldsymbol{\sigma}_t \cdot \boldsymbol{\epsilon}' + \boldsymbol{\mu}_t$ allows generating variants of the original input sequence, a process we refer to as variational sampling. Variational language modeling thus denotes the various ways of manipulations and computations in the latent space that are possible with the VAT's latent distributions. As examples, we discuss anecdotal results from variational sampling and interpolation.

Tables 1 and 2 illustrate the generating capabilities of the VAT with variational sampling. In the second line in Table 1, the VAT is able to reconstruct the original sequence (first line), which refers to decoding with $\boldsymbol{z}_t = \boldsymbol{\mu}_t$, i.e., without sampling. The following lines show exemplary variants of the input sentence obtained by randomizing the latent representations, displaying 12 variational samples. It is interesting that, while mostly maintaining the original sentence structure and original context, variants are found for almost all tokens. This gives an idea of the structure of the latent space and the contextuality of the representations. The approach is comparable to paraphrase generation, which is often obtained through back-translation [27,28] or by directly training on supervised data in the form of paraphrased sentence pairs [29].

Table 1. Variational sampling. Example sentence (in **bold**) used as the input to the VAT, with the decoded mean representation and samples from the latent distribution (in *italics*).

In this class, we will introduce several fundamental concepts.
In this class, we will introduce several fundamental concepts.
In this class, we will introduce several fundamental principles. *In this area, we will introduce several fundamental principles.* *In this process, we will introduce several educational concepts.* *In this class, I will introduce several fundamental principles.* *In this progress, we will introduce private fundamental concepts.* *In this class, we have received several fundamental concepts.* *In this class, we will introduce several public projects.* *In April 2013, we will introduce several fundamental issues.* *In this class, we will find several fundamental principles.* *In one class, she will introduce several fundamental concepts.*

Table 2. Variational sampling. Example sentence in (**bold**) that was used as the input to the VAT, with only the underlined token being sampled. Samples from the latent distribution in *italics*.

Generative models have shown great promise in modeling complex distributions.
advantages, results, achievements, quality, answers participation, moments, experience, passion, joy, benefits, interests

Table 1 illustrates variational sampling for a single token, where the input sequence (in bold) is reconstructed according to $z_t = \mu_t$, except for the underlined token, for which sampling is allowed. This setting is similar to a nearest neighbor search over a fixed corpus. The second line lists the obtained samples. By masking the underlined token, this approach is similar to gap filling [30,31]. Variational sampling, both for tokens and sequences, can be useful for generating paraphrases in tasks such as dialog generation or question answering.

We also experimented with latent space interpolation similar to [15,18,19]. We linearly interpolated the latent representations of two sentences (padded to the same length) with three intermediate steps. Decoding the intermediate representations results in the sentences illustrated in Table 3 for two examples. Observing a smooth interpolation trajectory as in the examples is not always the case. It is possible that after decoding, the intermediate steps are mapped to the same sentence, as seen for one sentence in the second example. This is especially true with the increasing size of the training data: the more training data, the more contextualized representations for the same word type that are mapped close to each other in latent space will exist.

Table 3. Interpolation. First and last sentence (**bold**) were given as the input to the VAT. Intermediate sentences were obtained by tokenwise linear interpolation between the two latent representations padded to the same length.

I want to talk to you.
I want to report.
I said:
She didn't want to say.
She didn't want to be with him.
He was silent for a long moment.
He was silent for a long moment.
He was my father.
It was my face.
It was my turn.

For further investigations on the distribution of the variational samples and their distance from the mean representations, the distances between latent vectors in terms of their accumulated elementwise differences (i.e., over all dimensions and vectors in the representation of the sentence) were assessed. Figure 4 illustrates the histogram of these elementwise distances for the mean vectors of the start and end sentence from the second example in Table 3 in blue. For comparison, ten random sentences were drawn from the original start sentence (with the same approach as in Table 1). The red histogram shows the distribution of their elementwise differences. In order to illustrate the effect of the regularization term, experiments with different scaling values s are shown to determine the "signal-to-noise ratio", where the differences between mean representations of start and end sentence are considered as the signal and those between the variational samples as noise. For $s = 0.4$, this ratio is 0.6 on average. A closer, dimensionwise look at all vectors (not included in Figure 4) revealed that the peak value is 2 and is achieved in only one outlier dimension: even this "feature" is thus considerably affected by the variational random noise. The histograms for $s = 0.9$ and $s = 0.1$ are in line with the findings in Figure 3a,b, where the representations either collapse into an identical representation for the sentences ($s = 0.9$) or the real distribution approximates the variational distribution more closely ($s = 0.1$).

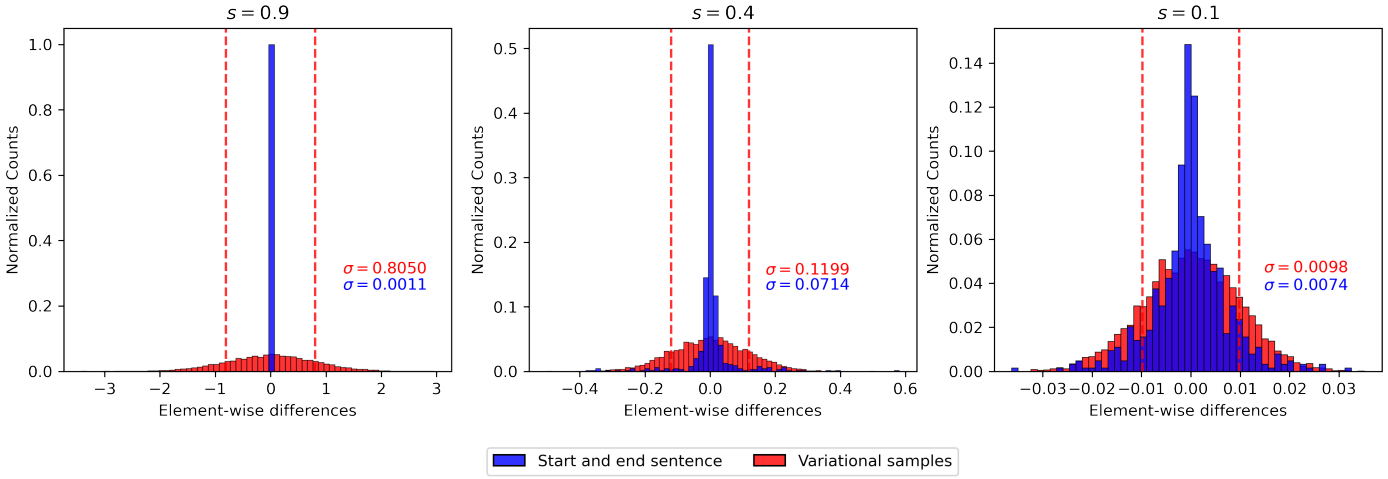


Figure 4. Histogram of the elementwise difference between token vectors from the start and end sentence of the interpolation example in Table 3 (blue) and from the start sentence and 10 random variational samples of this sentence (red) for different scaling values s. σ denotes the value of the standard deviation of the corresponding distributions.

7. Sentence Representations

For the construction of sentence representations, we compared two pooling operations (average, sum). Additionally, the start token ("CLS") representation was used as the embedding for the whole sentence, although this approach has been shown to be inferior to mean pooling in semantic similarity tasks for BERT [32]. We denote the different approaches as VAT-*mean*, VAT-*sum*, and VAT-*start*, respectively. We compared our model to the original BERT model (base) and, for a fair comparison in terms of model size, to a smaller variant with similar network dimensions as the VAT. BERT has $L = 12$ encoder layers, and the model and representation dimension is $d = 768$. The smaller BERT model (https://github.com/google-research/bert, accessed on 8 June 2022) [33], which we refer to as MiniBERT, has $d = 128$ and $L = 4$, the same size as the VAT. Both BERT and MiniBERT were trained on a much larger collection of datasets (Wikipedia, BookCorpus, CommonCrawl) than the VAT.

The sentence representations were tested using the SentEval [34] toolkit (https://github.com/facebookresearch/SentEval, accessed on 8 June 2022). The toolkit evaluates static sentence representations on two different classes of tasks: semantic similarity and sentence classification tasks. Table 4 lists the correlation values of the different sentence representation methods on the Semantic Textual Similarity (STS) tasks. For each pair of semantically similar sentences, the Spearman correlation rank between the cosine of their latent representations and a human-labeled gold standard (between 0 and 5) was computed (without fine-tuning or transfer learning). The correlation of VAT-based sentence representations is on par with BERT-based representations, which is surprising given the smaller model architecture and capacity of the VAT. For both VAT and BERT, start-token-based representations show less correlation than those obtained by average or sum pooling. Representations obtained from the MiniBERT model do not show any correlation at all, indicating that the BERT architecture requires a larger model size.

Table 4. Spearman rank correlation between the cosine similarity of sentence representations and gold label similarity for different Semantic Textual Similarity (STS) tasks without fine-tuning the underlying models to the target data.

	STS12	**STS13**	**STS14**	**STS15**	**STS16**	**STSb**	**SICK-R**	**Average**
				Published in [32]				
BERT-*start*	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
BERT-*mean*	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
				Our results				
MiniBERT-*start*	−2.19	0.87	1.27	0.81	−3.46	−2.41	0.61	−0.64
MiniBERT-*mean*	−1.14	−3.83	1.63	0.21	7.25	−5.3	−1.73	−0.42
MiniBERT-*sum*	−0.50	−1.54	0.60	−0.72	0.11	1.21	1.67	0.12
VAT-*start*	17.45	12.10	14.74	20.09	31.14	30.70	37.15	23.34
VAT-*mean*	45.60	45.43	52.07	55.45	58.13	50.83	47.08	50.66
VAT-*sum*	45.60	45.43	52.07	55.45	58.13	51.56	48.88	51.02

The classification tasks included in the SentEval toolkit comprise binary (MR, CR, SST2) and fine-grained (SST5) sentiment or polarity classification tasks, paraphrase detection (MRPC), natural language inference (SICK-E), and question-type classification (TREC) tasks. As paraphrase detection and natural language inference require the comparison of two sentences, the pair of input sentences was concatenated and separated by a special separation token to form a single-input vector. For the classification tasks, we compared the performance of the VAT to that of MiniBERT in two ways: in a feature-based approach and in a fine-tuning approach.

Both presented approaches are different from how state-of-the-art models are trained for transfer tasks. Given the smaller model capacity, we do not expect on-par performance, but rather want to better understand the kind of information stored in the representations

and the potential that comes with additional classification layers on top of the model. The comparison to MiniBERT serves as the baseline.

For the feature-based approach, the sentence representations are extracted from the pre-trained models and then passed to a single classification layer trained with an Adam optimizer [35]. Given the results in Table 5 (top), the sum of the individual token representations is best suited as a standalone feature for sentence classification. As VAT-*sum* outperforms VAT-*mean* on all tasks, the dimensions of the latent representations seem to be well disentangled. With the exception of the SST2 task, the start representation yields the lowest accuracy values, suggesting that the first representations do not capture the full context of a sentence. MiniBERT, which is designed for fine-tuning rather than feature extraction, mostly does not reach the performance of the VAT variants. The difference between MiniBERT and VAT is especially large for the TREC task.

For assessing the performance with a fine-tuning approach, we extended the underlying models by a single nonlinear dense classification layer and optimized with Adam. For the VAT, we only reused the encoder part to access the latent representations Z. For each task, we trained for five epochs and selected the best learning rate out of 3×10^{-4}, 1×10^{-4}, 5×10^{-5}, and 3×10^{-5} based on the validation set to be used on the test set. Note that the MiniBERT results can differ from the results in the original publication [33], as we did not further tune the model or the training process.

In Table 5 (bottom), we see a performance leap for all of the models compared to their feature-based results. While VAT-*sum* performed best for the feature-based approach, the method to produce sentence-level representations was not that decisive for the fine-tuning approach. The differences between *sum*, *mean*, and *start* were no longer as great as for the semantic similarity tasks, neither for MiniBERT nor for the VAT. It is interesting that MiniBERT is in the lead for sentiment classification tasks, whereas the VAT shows outstanding performance on the TREC (topic classification) task.

Table 5. Dev/test accuracies on the SentEval transfer classification tasks. A single dense layer with nonlinear activation and the Adam optimizer is used as the classification layer. **Bold** values indicate the best test result for each task in the feature-based and fine-tuning approach.

	MR	**CR**	**SST2**	**SST5**	**TREC**	**MRPC**	**SICK-E**
				Feature-Based Approach			
MiniBERT-start	50.8/50.1	63.9/63.8	51.2/49.9	27.7/24.6	23.7/18.2	67.8/66.5	56.4/56.7
MiniBERT-mean	50.9/50.1	61.6/60.9	53.0/50.3	24.4/25.1	22.9/18.2	65.5/65.9	55.6/55.9
MiniBERT-sum	51.9/49.6	62.2/59.7	51.4/51.5	25.8/23.0	23.0/20.8	64.8/57.6	48.4/49.7
VAT-start	60.5/59.6	63.8/63.8	**61.6/63.3**	29.4/29.3	35.1/40.4	67.5/66.5	56.4/56.7
VAT-mean	61.0/60.2	63.9/63.8	61.4/60.7	30.4/31.0	44.1/48.6	67.5/66.6	59.2/58.5
VAT-sum	**61.2/60.5**	**66.8/65.7**	61.0/62.6	**31.5/32.2**	**46.8/54.8**	**69.6/68.5**	**67.8/66.6**
				Fine-Tuning Approach			
MiniBERT -start	72.8/72.1	64.4/61.9	80.6/83.2	36.7/36.1	32.1/38.8	68.9/66.5	56.4/56.7
MiniBERT-mean	**76.0/75.6**	73.4/65.6	**81.0/83.3**	38.0/37.9	72.4/73.2	71.8/68.6	58.2/58.1
MiniBERT-sum	73.8/70.7	75.4/67.6	70.0/81.6	35.8/32.8	70.8/69.6	68.9/63.0	57.6/44.3
VAT-start	74.5/74.6	76.3/72.5	82.7/82.6	37.8/38.6	82.7/87.2	**70.0/69.9**	61.6/61.3
VAT-mean	73.8/74.1	75.0/72.5	81.7/82.4	**39.9/40.0**	84.1/85.2	68.9/63.7	**64.2/61.8**
VAT-sum	72.9/73.2	**78.6/74.5**	**82.6/83.3**	40.7/39.9	**84.7/88.2**	69.7/68.3	62.0/60.8

8. Discussion and Conclusions

The goal of the proposed VAT model is to obtain isotropic representations mapped to a smooth and complete latent space. The hyperparameter and scaling factor s is directly optimized to meet these criteria, thus reducing the effects of overfitting. The VAT is able to produce coherent sequences through manipulations directly in latent space. However, optimizing $\mathcal{L}_{REG}$ comes at the cost of $\mathcal{L}_{REC}$. Perfect reconstruction of the original input is no longer possible at all times. However, this ability to produce variational samples will be especially useful in language generation tasks such as question answering. Conversational agents could benefit from paraphrases generated by variational sampling far as it would produce more nuanced and varied replies instead of deterministic answers.

Reducing the representations to a single sentence-level vector preserves contextual information. Semantic similarity tasks demonstrate that the smaller model capacity of the VAT is sufficient to capture semantic information that is on a par with larger BERT architectures. However, models explicitly trained to improve the performance on semantic similarity tasks are out of reach.

The VAT is able to produce more robust standalone sentence-level features when compared with MiniBERT for the feature-based classification approach. VAT outperforms the similar-sized MiniBERT also when being fine-tuned on sentence classification tasks, but is far from the state-of-the-art performance known from BERT-sized models with more sophisticated training procedures. Interestingly, the VAT shows its peak performance for the topic classification task TREC. In our setup, the VAT captures topic-related information even better than sentiment. A possible explanation could be that topic-related information is encoded in several token representations of a sentence. Sentiment might be inherent in less tokens and, thus, be underrepresented in the aggregated sentence representation.

Tasks involving the comparison of two sentences (MRPC, SICK-E), represented by a single concatenated vector, cannot be successfully solved by either the VAT or MiniBERT. Some model variants (MiniBERT included) only learn to predict the most frequent class. The reason could be the smaller model sizes, which are incapable of capturing the information of two concatenated sentences, as especially the VAT was never trained on such a setting. Instead of a concatenated vector, the comparison task could be solved using a Siamese network architecture, similar to Deudon et al. [10].

The comparison to MiniBERT leaves us optimistic that the results could be successfully extrapolated to larger models. We expect the performance on downstream tasks to improve while still maintaining the desired characteristics of the latent representations supporting semantic reasoning through vector space arithmetic. The optimization of the regularization loss at the token level could even allow the training of a multilingual model or a translation model, given sufficient training data.

Author Contributions: Conceptualization, C.F. and S.W.; methodology, C.F. and S.W.; software, C.F.; validation, C.F.; formal analysis, C.F. and S.W.; investigation, C.F.; data curation, C.F.; writing–original draft preparation, C.F.; writing–review and editing, C.F. and S.W.; visualization, C.F.; supervision, C.F. and S.W.; project administration, C.F. and S.W.; funding acquisition, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This project is partially funded by the Science and Innovation Strategy Salzburg (WISS 2025) project "IDA-Lab Salzburg", Grant Number 20102-F1901166-KZP.

Data Availability Statement: The data used in this study are openly available: WMT19 [24] is available through the tensorflow library at https://www.tensorflow.org/datasets/catalog/wmt19_translate (accessed on 8 June 2022). SentEval [34] is available from Facebook Research at https://github.com/facebookresearch/SentEval (accessed on 8 June 2022).

Acknowledgments: The authors want to thank Salzburg University of Applied Sciences for its doctoral support program, which facilitated the work on the publication.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
2. Ethayarajh, K. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. 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), Hong Kong, China, 3–7 November 2019; pp. 55–65.
3. Gao, J.; He, D.; Tan, X.; Qin, T.; Wang, L.; Liu, T. Representation Degeneration Problem in Training Natural Language Generation Models. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
4. Wang, L.; Huang, J.; Huang, K.; Hu, Z.; Wang, G.; Gu, Q. Improving Neural Language Generation with Spectrum Control. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.

5. Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; Li, L. On the Sentence Embeddings from Pre-trained Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 9119–9130.
6. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
7. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
8. Gururangan, S.; Dang, T.; Card, D.; Smith, N.A. Variational Pretraining for Semi-supervised Text Classification. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5880–5894.
9. Mahabadi, R.K.; Belinkov, Y.; Henderson, J. Variational Information Bottleneck for Effective Low-Resource Fine-Tuning. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
10. Deudon, M. Learning Semantic Similarity in a Continuous Space. In *Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2018; Volume 31.
11. Zhao, T.; Lee, K.; Eskenazi, M. Unsupervised Discrete Sentence Representation Learning for Interpretable Neural Dialog Generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 1098–1107.
12. Miao, Y.; Yu, L.; Blunsom, P. Neural Variational Inference for Text Processing. In Proceedings of the 33rd International Conference on Machine Learning (ICML'16), New York, NY, USA, 20–22 June 2016; Volume 48, pp. 1727–1736.
13. Wang, T.; Wan, X. T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion. In Proceedings of the IJCAI, Macao, China, 10–16 August 2019; pp. 5233–5239.
14. Shu, R.; Lee, J.; Nakayama, H.; Cho, K. Latent-variable Non-autoregressive Neural Machine Translation with Deterministic Inference using a Delta Posterior. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34; pp. 8846–8853.
15. Bowman, S.R.; Vilnis, L.; Vinyals, O.; Dai, A.; Jozefowicz, R.; Bengio, S. Generating Sentences from a Continuous Space. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 10–21.
16. Fang, L.; Li, C.; Gao, J.; Dong, W.; Chen, C. Implicit Deep Latent Variable Models for Text Generation. 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), Hong Kong, China, 3–7 November 2019; pp. 3946–3956.
17. Yang, Z.; Hu, Z.; Salakhutdinov, R.; Berg-Kirkpatrick, T. Improved Variational Autoencoders for Text Modeling Using Dilated Convolutions. In Proceedings of the 34th International Conference on Machine Learning (ICML'17), Sydney, NSW, Australia, 6–11 August 2017; Volume 70, pp. 3881–3890.
18. Liu, D.; Liu, G. A Transformer-Based Variational Autoencoder for Sentence Generation. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–7.
19. Li, C.; Gao, X.; Li, Y.; Peng, B.; Li, X.; Zhang, Y.; Gao, J. Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual, 16–20 November 2020; pp. 4678–4699.
20. Li, R.; Li, X.; Chen, G.; Lin, C. Improving Variational Autoencoder for Text Modelling with Timestep-Wise Regularisation. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 2381–2397.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*; Curran Associates, Inc.: New York, NY, USA, 2017; pp. 5998–6008.
22. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
23. Odaibo, S. Tutorial: Deriving the Standard Variational Autoencoder (VAE) Loss Function. *arXiv* **2019**. arXiv:1907.08956
24. Barrault, L.; Bojar, O.; Costa-Jussà, M.; Federmann, C.; Fishel, M.; Graham, Y.; Haddow, B.; Huck, M.; Koehn, P.; Malmasi, S.; et al. Findings of the 2019 Conference on Machine Translation (WMT19). In Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1), Florence, Italy, 1–2 August 2019; pp. 1–61.
25. Agirre, E.; Cer, D.; Diab, M.; Gonzalez-Agirre, A. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. *SEM 2012: The First Joint Conference on Lexical and Computational Semantics—Volume 2. In Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), Montréal, QC, Canada, 7–8 June 2012; pp. 385–393.
26. Lucas, J.; Tucker, G.; Grosse, R.; Norouzi, M. Understanding Posterior Collapse in Generative Latent Variable Models. In Proceedings of the International Conference on Learning Representations, DeepGenStruct Workshop, New Orleans, LA, USA, 6–9 May 2019.
27. Sennrich, R.; Haddow, B.; Birch, A. Improving Neural Machine Translation Models with Monolingual Data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 86–96.
28. Wieting, J.; Gimpel, K. ParaNMT-50M: Pushing the Limits of Paraphrastic Sentence Embeddings with Millions of Machine Translations. In Proceedings of the 56th Annual Meeting of the Association for Computational LinguisticsMelbourne, Australia, 15–20 July 2018; pp. 451–462.

29. Gupta, A.; Agarwal, A.; Singh, P.; Rai, P. A deep generative framework for paraphrase generation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
30. Donahue, C.; Lee, M.; Liang, P. Enabling Language Models to Fill in the Blanks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 2492–2501.
31. Wu, X.; Zhang, T.; Zang, L.; Han, J.; Hu, S. Mask and Infill: Applying Masked Language Model for Sentiment Transfer. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria, Macao, China, 10–16 August 2019; pp. 5271–5277.
32. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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), Hong Kong, China, 3–7 November 2019; pp. 3982–3992.
33. Turc, I.; Chang, M.W.; Lee, K.; Toutanova, K. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv* **2019**. arXiv:1908.08962v2
34. Conneau, A.; Kiela, D. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018); European Language Resources Association (ELRA), Miyazaki, Japan, 7–12 May 2018.
35. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.