Language Modelling using Latent Diffusion Models

Stanford CS224N Custom Project

Ryan Po
Department of Electrical Engineering
Stanford University
rlpo@stanford.edu

Sebastian Charmot
Department of Statistics
Stanford University
scharmot@stanford.edu

Abstract

Denoising diffusion probabilistic models (DDPMs) [1] have seen explosive growth in the past few years, especially in the domain of image synthesis. Due to the nature of DDPMs, these methods are especially well suited for modelling continuous data. Despite recent work showing that DDPMs outperform other state-of-the-art methods in image synthesis [2], there is limited work applying diffusion methods to the domain of language modelling. We propose a method that uses latent diffusion models for language modelling. Using a sentence autoencoder architecture [3], we map sentences to a continuous latent space and perform diffusion over the latent space.

1 Key Information to include

- Mentor: Lisa Li
- External Collaborators (if you have any): None
- Sharing project: N/A

2 Introduction

Language modeling is a crucial task in natural language processing research, especially with the recent increase in attention to the field due to the incredible performance from models trained on internet scale data. Recent state-of-the-art language modeling methods have been dominated by transformer-based methods, with representative works such as GPT [4], BERT [5] and RoBERTa [6]. The field of generative AI has seen a recent resurgence with a series of work proving that diffusion models have the ability to outperform other state-of-the-art architectures in the domain of image synthesis [2]. However, there is currently limited work in applying advances in diffusion models to the domain of language modeling.

Applying diffusion models to language modeling is challenging due to the discrete nature of language data. Although DDPMs have shown to work well on image data, the distribution of image data and language data is fundamentally different. Current representative works in applying diffusion models to the domain of language modeling include Diffusion-LM [7] and DiffuSeq [8]. Both methods apply diffusion models at the level of word embeddings. Since the word embedding space is continuous, training diffusion models on this data leads to great results. However, in order to map the diffused embeddings back to words, a rounding procedure must be performed, as there is no gaurantee that the diffusion samples are going to be identical to word embeddings.

Recent work has shown that latent diffusion models can lead to faster and better diffusion sampling [9]. In our method, we take inspiration from these methods and apply latent diffusion to language modeling. We first use a VAE in order to encode sentences into a continuous latent space using the method proposed by Bowrman et al. [3] and train a diffusion model to learn the data distribution of the resulting latents.

3 Related work

Diffusion models. Recently, explosive advances in diffusion models have led to a shift in generative models utilizing denoising diffusion probabilistic models (DDPMs) as their backbone. Representative work by Ho et al [1] established the mathematical foundation of such models in the context of image generation. Later work showed that DDPMs have the ability to outperform state-of-the-art methods in image synthesis tasks (most state-of-the-art techniques up until then utilized GANs) [2]. With this in mind, diffusion models have shown a great deal of potential in the space of generative modelling. Qualitatively, DDPMs have shown the ability to generate samples with higher diversity, in contrast to GANs which empirically suffer from mode-collapse. Our method aims to leverage the recent advances in diffusion models for the task of language modelling.

Variational autoencoders. Another popular generative modelling method is the variational autoencoder (VAE). VAEs are a type of deep generative model that has shown effectiveness in learning a complex probability distribution and generating high-fidelity samples. First proposed by Kingma and Welling [10], VAEs are autoencoder architectures trained using variational inference. Prior work has shown that VAE effectively maps a data distribution to a lower-dimensional regularized probability distribution (usually a Gaussian). However, VAEs have shown to suffer from posterior-collapse, a phenomenon that occurs when the input signal is not being properly represented by the encoded latent. In order to combat this, prior work [11] has shown that adding a weighting term to the KL-divergence loss allows the latent distribution to train freely. In our work, we also explore the use of different β values during training, and even a dynamic scheduler for annealing the values of β.

Latent diffusion models. Although diffusion models have shown great potential for generative modeling, sampling diffusion models are a painfully slow process due to the method's nature. While other generative methods such as VAEs and GANs only require a single pass through a network with a randomly generated latent code, sampling a DDPM requires many passes through the denoising network before a reasonable output is formed. In order to tackle this problem, recent work has shown that training a DDPM to learn the lower dimension latent distribution of an autoencoder allows for much more efficient training and inference while also maintaining high-fidelity generations. We also choose to leverage latent diffusion models in our method. However, the motivation behind our choice does not stem from efficiency, but the fact that the latent space of a VAE is continuous. Language data is often discrete, and DDPMs empirically struggle with discrete data, and have only shown to perform well for continuous data. Our method leverages an autoencoder to first map the data to a continuous space, which can then act as training data for the generative DDPM.

Language diffusion models. Recent work has applied diffusion models in the domain of language modeling, and in some cases even achieving state-of-the-art results. Representative work in the field by Li et al. [7] has shown that applying diffusion models at the word embeddings level leads to impressive results. Another work by Gong et al. [8] also explores diffusion at the embedding level. Although word embeddings lie in a continuous space, mapping from DDPM-generated samples to a word embedding requires rounding. For our method we wish to look at diffusion at the sentence level, which means that we do not have to do any rounding operations during training, and the diffusion model output latents can be directly decoded.

4 Approach

Applying diffusion models to language modelling is inherently challenging due to the discrete nature of language data. Therefore, our approach aims at first mapping our training data into a continuous space that is better suited for diffusion modelling. Our method takes inspiration from the popular StableDiffusion [9] latent diffusion method, where a pretrained autoencoder architecture maps images to a latent space, from which diffusion modelling is then performed. In our case, we use a sentence variational autoencoder (VAE) as proposed by [3] in order to map a set of sentences into a continuous latent space. After training the VAE on a given dataset, we encode sentences from this dataset into the latent space using the trained encoder. This set of latent vectors will act as the target distribution for our diffusion model.

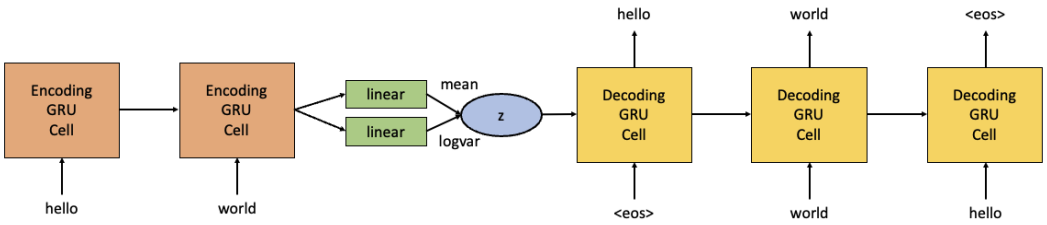

Figure 1: Sentence VAE overview.

4.1 Diffusion model preliminaries

Recent work has shown that diffusion models can achieve state-of-the-art quality for image generation tasks [2]. Specifically, Denoising Diffusion Probabilistic Models (DDPMs) implement image synthesis as a denoising process. DDPMs begin from sampled Gaussian noise x_T and applies T denoising steps to create a final image x_0. The forward diffusion process q is modelled as a Markov chain that gradually adds Gaussian noise to a ground truth image according to a predetermined variance schedule $\beta_1, \beta_2, \ldots, \beta_T$

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}\right) \tag{1}$$

The goal of DDPMs is to train a diffusion model to revert the forward process. Specifically, a function approximator $\boldsymbol{\epsilon}_\phi$ is trained to predict the noise $\boldsymbol{\epsilon}$ contained in a noisy image x_t at step t. $\boldsymbol{\epsilon}_\phi$ is typically represented as a convolutional neural network characterised by its parameters ϕ. Most successful models train their models using a simplified variant of the variational lower bound on the data distribution:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{t,x,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\phi\left(x_t, t\right)\right\|^2\right] \tag{2}$$

with t uniformly sampled from $\{1, \ldots, T\}$. The resulting update step for obtaining a sample for x_{t-1} from x_t is then

$$x_{t-1} = x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\phi(x_t, t) + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\mathcal{N}(0, \mathbf{I}) \tag{3}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$, $\alpha_t = 1 - \beta_t$.

Following previous work, we use a standard U-Net to represent ϵ_ϕ. In our case, **we wrote a custom diffusion pipeline**, that involves modelling the forward process and a training objective that aims to model the reverse denoising process.

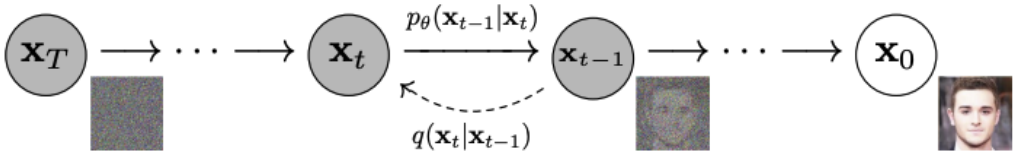

Figure 2: Diffusion model overview. Figure from Ho et al. [1]

4.2 Diffusion for language modelling

Recent work has shown the diffusion models can be adapted for language modelling. In fact, representative work done by Li et. al [7] has shown that diffusion models can outperform previous state-of-the-art methods in conditional text generation. The authors chose to directly diffuse word

vectors in their approach, and their key to overcoming the discrete nature of language data was to clamp and round the diffused embedding vectors to the nearest word embedding to map the continuous space into the discrete vocabulary space.

Our method differs from the method proposed by Li et. al [7], in that we use the VAE to map an entire sentence to a continuous space and directly perform diffusion on this latent space. This means that during the diffusion process, we do not need to do any rounding, instead, the raw output from the diffusion model is the latent vector being passed into the trained decoder.

4.3 Baselines

Following the work in [3], we use the standard VAE model proposed in their paper as our primary baseline. This is a reasonable baseline since this project aims at investigating whether the diffusion model is able to leverage the latent space created by an auto-encoder. Note that this baseline produces samples by sampling a Gaussian distribution and using them as latents for sentence generation. We compare our methods that use a diffusion model for sampling latents instead.

5 Experiments

5.1 Data

For our experiments, we use the Penn TreeBank [12] dataset which consists primarily of articles from the Wall Street Journal. We made a modification to the dataset by removing all consecutive <UNK> and N tokens in the dataset. We found that without performing this augmentation to the data, the trained model has a tendency to generating many consecutive <UNK> tokens during the decoding stage.

We train our VAE with the train split which contains approximately 40000 sample sentences. For each trained VAE, we then freeze the encoder and encode all the train sentences to obtain a latent mapping to each sentence in the dataset. These latents are then passed into the diffusion model for training.

Latent vectors serving as diffusion model data is normalized to have a mean of 0.5 and variance of 0.5, which is consistent with most diffusion model training regimes.

5.2 Evaluation methods

We evaluate our method for language modeling compared against the stock VAE method using MAUVE score, perplexity, and human evaluation.

MAUVE Score. The MAUVE Score is a metric for open-ended text generation that compares the learnt distribution from a text generation model to the distribution of human-written text using divergence frontiers computed in a quantized embedding space [13]. The MAUVE score is bounded between 0 and 1 where a score of 1 indicates perfect alignment between the text generation model's distribution and the distribution of human-written text.

Perplexity (PPL). Perplexity measures how likely the generated samples are according to an autoregressive language model. We use GPT2-Large to compute perplexity [14]. Perplexity is formally defined as the exponentiated average negative log-likelihood of a sequence. Therefore, if we have a tokenized sequence $X = (x_0, x_1, \ldots, x_t)$, then the perplexity of X is,

$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta\left(x_i \mid x_{<i}\right)\right\}$$

where $\log p_\theta\left(x_i \mid x_{<i}\right)$ is the log-likelihood of the i^{th} token conditioned on the preceding tokens $x_{<i}$ according to our model. Generally speaking, a lower PPL value indicates better performance for a language model.

Human Evaluation. Human evaluation is the "gold standard" for assessing the performance of a language model. To perform human evaluation, we asked 4 individuals to rate the quality of various

Model Type	Model Parameters		Evaluation Metrics			Mean Length
	Latent Size	Beta	PPL ↓	MAUVE ↑	Human Eval ↑	
Reference	-	-	349.3	0.946	8.05	20.91
Baseline VAE	16	Anneal	264.9	0.114	4.62	18.08
	64		**209.9**	0.095	4.71	18.96
	256		216.1	0.058	4.16	16.53
Diffusion Model	16	1e-5	409.8	0.236	**5.81**	24.99
		1e-4	402.7	0.271	4.23	22.12
		1e-3	428.2	**0.291**	5.18	22.32
		Anneal	245.4	0.137	5.03	19.14
	64	1e-5	723.6	0.151	4.41	28.30
		1e-4	703.7	0.166	5.24	22.85
		1e-3	749.2	0.226	3.41	24.71
		Anneal	262.1	0.094	3.06	20.44
	256	1e-5	735.6	0.164	4.13	26.75
		1e-4	759.4	0.176	4.02	26.67
		1e-3	720.6	0.241	3.62	24.34
		Anneal	255.7	0.075	3.59	17.59

Table 1: Comparison of MAUVE, perplexity scores from our model vs. the VAE

outputs from 1 - 10 based on a how fluent, coherent, and grammatically correct each output was. Then, we average the score of the four individuals to create one final human evaluation score per output. Due to the cost of human evaluation, fifteen random samples were selected from each model.

5.3 Experimental details

We trained the VAE on the PTB dataset and trained a diffusion model on the resulting latents computed from sentences in the training set. Unlike the original work, we chose to use GRUs instead of LSTMs for the encoder and decoder of the VAE. We trained our model with a learning rate of 1e-3.

For the diffusion model, we decided on a standard U-Net, with 3 down/upsampling steps trained at a learning rate of 1e-3.

To create a reference for our metrics, we calculate perplexity and MAUVE score on PTB. The reference perplexity is calculated by averaging the perplexity of each sentence within PTB's test set. The reference MAUVE score is calculated from a comparison of 1000 randomly sampled sentences from PTB's train set and 1000 randomly sampled sentences from the test set.

5.4 Results

The results of our models and hyper-parameters are displayed in Table 1. The model with the best PPL performance was the baseline VAE with a latent size of 64. The model with the best MAUVE performance was our diffusion based model with a latent size of 16 and β of 1e-3. The model with the best human evaluation performace was our diffusion based model with a latent size of 16 and β of 1e-5.

We can also qualitatively review some of the generations from each model. In Table 2, we show samples from PTB's test set, generations from the baseline VAE using a latent size of 64, and generations from our diffusion based model with latent size 16 and β of 1e-3.

6 Analysis

There are several interesting observations in our results. First, the annealed β values tend to perform better in terms of PPL for our diffusion based models. The β value is used to balance the reconstruction

Reference Text (Test Set)	Generations from Baseline VAE	Generations from Diffusion
speculators are calling for a degree of liquidity that is not there in the market	but the company said it wouldn't be able to sell three shares	nonetheless the company has a rationale for the past two weeks to discuss the company's future problems
futures traders say the s&p was <unk> that the dow could fall as much as N points	the best way to keep the <unk> of the <unk> is in the best way	in an interview with the securities exchange officials said they didn't want to take advantage of the currency
canada savings bonds are major government instruments for meeting its financial requirements	the issue is a sign of a big wall street journal's very very painful securities firms	the big board's efforts to revive the stock market's third-quarter earnings
japan's opposition socialist party denied that its legislators had been <unk> by <unk> owners	but the pilots have agreed to pay the company's suit in the suit filed suit against bellsouth and other bidders	bankers trust that its offer to buy back from n but the company's problems weren't fully accurate by mr . edelman

Table 2: Generations from the stock VAE and our latent diffusion model.

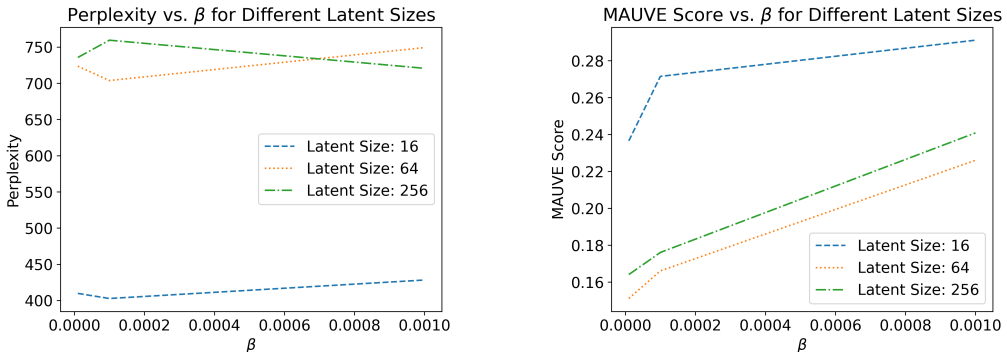

Figure 3: Ablations on β for both Perplexity and MAUVE Score

error and the KL divergence term in the VAE objective function. By annealing β, the diffusion model should be learning a disentangled representation of the data in a more stable manner. This seems to have an advantage in terms of PPL over setting a discrete value to the importance of the KL divergence term relative to the reconstruction error.

The second interesting behavior from our diffusion models is that the MAUVE score tends to be higher for fixed β values as apposed to annealing. It might be the case that a fixed value of beta strikes a better balance between the reconstruction error and the KL divergence term, which results in a better representation of the semantic similarity between words.

To better understand how our β and latent size parameters affected the performance of our diffusion based models, we performed an ablation on each.

6.1 Ablations on β

We illustrate the effect on PPL and MAUVE by varying β in Figure 3. Generally speaking, varying β doesn't seem to cause significant change in PPL. Varying β does seem to have a much greater effect on the MAUVE score given a fixed latent size. A higher value of β more strictly enforces the KL divergence term, which can create a more disentangled latent space representation of the data since

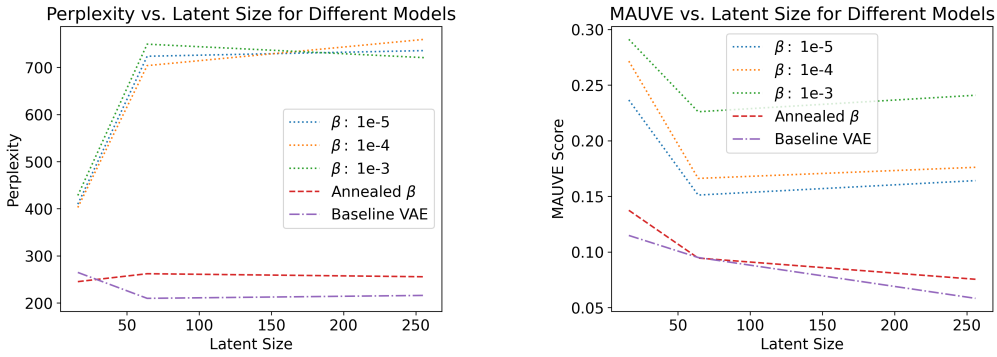

Figure 4: Ablations on Latent Size for both Perplexity and MAUVE Score

each dimension of the latent space corresponds to a specific, independent feature or attribute of the data. This disentangled representation can help the model to capture more abstract semantic concepts and capture the semantic similarity between words, which can lead to a better MAUVE score. Note that the original sentence VAE paper [3] uses an annealed β schedule, where they gradually increase β during training. We implement this annealed β schedule and compare it with a constant β. Please refer to the original paper by Bowman et al. [3] for full implementation details.

6.2 Ablations on Latent Size

We investigated the effect of varying the latent size on PPL and MAUVE. For fixed β models, a smaller latent size seems to correspond to lower PPL values and higher MAUVE scores. For annealed β, the latent size does not seem to change the PPL values significantly. It does seem to be inversely proportional to the MAUVE score, however. Generally speaking, smaller latent sizes tend to correspond to stronger PPL and MAUVE performance metrics. The regularization from a smaller latent size could help with the model's generalizability.

6.3 Length of Generated Sequences

Previous research has shown that as the length of generated text increases, there is an expected decrease in quality of PPL and MAUVE [13]. In Table 1, we include the average lengths of generated sequences from our various models. We see that our diffusion based models produce longer sequences on average compared to the baseline, which makes it harder for our diffusion based models to perform equally as well on PPL and MAUVE score. Although that is the case, our diffusion based models are still able to outperform the baseline in terms of MAUVE and human evaluation.

7 Conclusion

In this paper, we devised a latent diffusion model for language modelling using a sentence autoencoder architecture. We were able to map sentences to a continuous latent space and perform diffusion over the latent space to generate sequences of text. Our results were able to outperform baseline VAE's on MAUVE score and human evaluation. Due to computing constraints, we were only able to perform coarse ablations on β and latent size hyper-parameters. It would be interesting to do a more fine-grained ablation study to understand the relationship between the hyper-parameters and metrics better. It would also be interesting to test other datasets. The PTB dataset contains quite a lot of financial jargon, and its reference text has a relatively high perplexity score. Testing larger datasets with lower diversity could be more beneficial for diffusion based approaches. It could also be worthwhile to test consistent architectures for the baseline as well as diffusion models to provide for a better direct comparison.

References

[1] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

[2] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021.

[3] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Conference on Computational Natural Language Learning*, 2015.

[4] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[7] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-lm improves controllable text generation. *ArXiv*, abs/2205.14217, 2022.

[8] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *ArXiv*, abs/2210.08933, 2022.

[9] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021.

[10] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[11] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

[12] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Comput. Linguistics*, 19:313–330, 1993.

[13] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4816–4828. Curran Associates, Inc., 2021.

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