Supplementary Materials for the Paper: BioVAE: a pre-trained latent variable language model for biomedical text mining

A BioVAE Model

We present the overview of our BioVAE model in Figure 1.

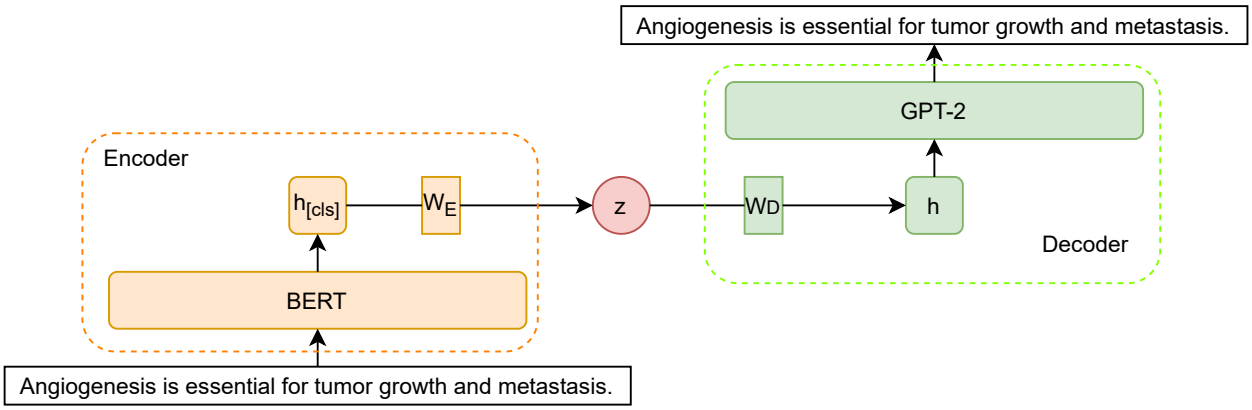

Figure 1: Overview of our BioVAE model. The last-layer hidden state $h_{[CLS]}$ from BERT is used as the sentence-level representation. The latent representation z is constructed via the weight matrix W_E. For decoder, the latent representation z is passed through the weight matrix W_D to construct the input embedding h for GPT-2 to reconstruct the input sequence.

In the original OPTIMUS model [10], BERT weights are initialized by the pre-trained BERT-based model [4], which is trained on general corpora (800M words of the BookCorpus and 2,500M words of English Wikipedia), and GPT-2 weights are initialized by the pre-trained GPT-2 model [17], which is trained on 40 GB of text from WebText created by scraping web pages.

In training our BioVAE model for biomedical domain, instead of using the BERT-based model, we initialized BERT weights by the pre-trained SciBERT model [1], which is trained on 3.17B tokens from 1.14M papers from Semantic Scholar containing 82% from the broad biomedical domain. We used the same pre-trained GPT-2 model [17] since there is no such model trained on biomedical domain available.

In the evaluation of our BioVAE model on text mining tasks such as NER, REL, QA, we directly use the pre-trained BioVAE encoder's BERT for fine-tuning.

B Generated samples

We compare sentences generated by our BioVAE and OPTIMUS models in Table 1.

Table 1: Reconstruction samples generated by OPTIMUS and our BioVAE. (Perplexity: lower is better.)

No.	Model	Texts	Perplexity
1	Input	Bevacizumab was discontinued in 2 patients because of nonfatal intracranial **bleeding**.	1.000
	BioVAE	Bevacizumab was discontinued in 2 patients because of nonfatal intracranial **hemorrhage**.	1.129
	OPTIMUS	Dydrogesterone has not been approved for use in children under 12 years of age.	3.070
2	Input	TUNEL assay and **microvessel** density was assessed to evaluate **apoptosis and angiogenesis.**	1.000
	BioVAE	TUNEL assay and **microvascular** density was assessed to evaluate **angiogenesis and apoptosis.**	1.120
	OPTIMUS	Virologic tests performed on the cells were performed by the U.S. Department of Health and Human Services.	4.609
3	Input	This implies that enhancing Nrf2 activity is a promising **method** for **thwarting** cancer.	1.000
	BioVAE	This implies that enhancing Nrf2 activity is a promising **tactic** for **preventing** cancer.	1.362
	OPTIMUS	This may lead to the development of a therapeutic agent that can be used in the treatment of neurodegenerative diseases.	3.504
4	Input	Spontaneous **metastasis** indicates a possible reverse correlation.	1.000
	BioVAE	Spontaneous **transit** indicates a possible reverse correlation.	2.075
	OPTIMUS	In contrast, the fluoxetine receptor does not appear to be affected by the presence of fluoxetine.	3.456
5	Input	Angiogenesis is **essential** for tumor growth and metastasis.	1.000
	BioVAE	Angiogenesis is **crucial** for tumor growth and metastasis.	2.283
	OPTIMUS	Transplantation of progesterone is an important factor in the development of cancer.	3.753

C Discussion on results

SciBERT Baseline Our baseline is SciBERT [1]. As presented in the paper (Approach), and in Appendix A, our model consists of two parts: an encoder and a decoder. BERT is used for the encoder, and GPT-2 is used for the decoder. In OPTIMUS [10], BERT is initialized by the general domain pre-trained BERT-based model [4]. Instead, in our work, since we aim at biomedical domain, we initialized BERT by the pre-trained biomedical SciBERT model [1]. During training the BioVAE model, BERT's weights are also updated. Therefore, we would like to investigate whether training our VAE-based language model (BioVAE) can improve the SciBERT. In other words, we compare the BERT model before training BioVAE (the initialized pre-trained SciBERT model) with the BERT model after training BioVAE (our BioVAE's encoder). From the results in Table 2, our BioVAE outperforms the SciBERT in all of the tasks.

Comparison with other pre-trained BERT models We also compare our BioVAE with the other biomedical pre-trained BERT models: BioBERT [9] and PubMedBERT [6]. From the results in Table 2, our BioVAE also outperforms the BioBERT in all of the tasks. PubMedBERT obtains better scores than BioVAE on REL and QA tasks, but lower performance on the NER tasks.

Discussion on PubMedBERT results It is noted that the BioBERT and SciBERT scores reported in PubMedBERT paper [6] are different from the original scores in the BioBERT and SciBERT papers because of some changes made by the PubMedBERT's paper in training settings. Additionally, PubMedBERT's evaluation scripts are not publicly available. Therefore, all scores we report here are based on the same settings and evaluation scripts provided by the SciBERT [1], which are publicly available at SciBERT repository[1] (for NER and REL tasks), and BioBERT scripts [9][2] for QA tasks. Based on these evaluation scripts, we replicated the original scores of SciBERT and BioBERT reported in their papers. Since PubMedBERT's evaluation scripts are not publicly available, the PubMedBERT scores we report here are also based on our runs using the SciBERT and BioBERT evaluation scripts. In this work we used SciBERT as our baseline to initialize BERT in the BioVAE. In future work, we plan to alternatively initialize BERT in BioVAE by using the pre-trained PubMedBERT model to train the BioVAE.

Table 2: Results on the text mining test sets. The best scores are in bold, and the scores outperforming the SciBERT baseline are underlined. We report macro F1 scores for NER, micro F1 for REL, and accuracy for QA. (d_z: latent size)

Model	NER			REL	QA
	BC5CDR	NCBI	JNLPBA		
PubMedBERT [6]	87.27	79.96	71.82	**85.47**	**75.00**
BioBERT [9]	88.85	89.36	77.59	76.68	69.29
SCIBERT [1]	90.01	88.57	77.28	83.64	72.14
BioVAE ($\beta = 0.0, d_z = 32$)	89.85	88.85	**77.82**	83.68	72.86
BioVAE ($\beta = 0.0, d_z = 768$)	90.10	88.12	77.69	83.05	72.14
BioVAE ($\beta = 0.5, d_z = 32$)	89.69	89.80	77.66	83.54	72.14
BioVAE ($\beta = 0.5, d_z = 768$)	**90.18**	**90.12**	77.57	84.49	72.86

D BERT-based pre-trained language models

We discuss here the background of BERT-based pre-trained language models (PLMs) presented in Introduction. BERT stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers [4]. This is a well-known and powerful language representation model. BERT advances the state-of-the-art (SOTA) on eleven natural language processing (NLP) tasks such as question answering, sentiment analysis, language inference etc. BERT-based is a neural-based, large pre-trained PLMs with 110M parameters. The

[1] https://github.com/allenai/scibert

[2] https://github.com/dmis-lab/biobert-pytorch/tree/master/question-answering

model is trained on 800M words of the BooksCorpus and 2,500M words of English Wikipedia, and freely available.[3] The pre-trained BERT-based model can be fine-tuned for a wide range of tasks such as named entity recognition, question answering, language inference, etc and achieved SOTA performances. It does not require substantial modifications and reduces the need for many heavily-engineered task-specific architectures.

E Backgrounds

In this part, we explain in more details about the backgrounds related to pre-trained language models, deep generative models, and variational autoencoders presented in Introduction.

E.1 Pre-trained language models (PLMs)

Language representation Neural networks have been widely applied to solve various NLP tasks. One of the foundation and important tasks is language representation. Neural models represent language semantic and syntactic features by using low-dimensional vectors (distributed representation) [15] (also called continuous representations or embeddings). Components of language such as words, phrases, or sentences, etc can be represented (or embedded) into vector space models, which allow to capture dependencies or relationships between the components.

Language models Language model is a basis of various NLP tasks. A simple concept of language model is training to predict the next word or words in a text based on the preceding words. Neural network language models use word embeddings or vector representations to make the predictions.

Word representations The early works of language representation task are related to word representations to learn vector representations of words from large amounts of unstructured text data. A word embedding (or word vector) is a learned representation in which words with similar meaning have a similar representation. The word representations from the learned vectors explicitly encode many linguistic regularities and patterns [12, 13]. One of the powerful methods is Skip-gram [12], which finds word representations that are useful for predicting the surrounding words in a sentence or a document. Another well-known word embedding model is GloVe [13]. Training from large scale unlabeled text can help word vectors to capture syntactic and semantic information of words. Pre-trained word embeddings become an essential component in many SOTA NLP architectures [7, 21].

Language model pre-training One problem of previous word representations methods is that learning word vectors only allows a single context-independent representation for each word. In order to overcome the problem, methods of learning embeddings from language models have been proposed. A model called ELMo [14] learns word representations which are functions of the entire input sentence. Each token's representation is a combination of context-sensitive features from a left-to-right and right-to-left language models. ELMo is trained on the 1B Word Benchmark, and advances SOTA on several NLP benchmarks [14, 20].

Feature-based and Fine-tuning After language models are trained on large amounts of unstructured text to learn representations to form pre-trained language models (PLMs), they can be applied to downstream tasks without training the models on the large datasets from scratch. The pre-trained models such as GloVe [13], ELMo [14] are used as additional features in existing task-specific architectures, which we call *feature-based* PLMs. Another strategy to applied PLMs for downstream tasks is called *fine-tuning*, in which PLMs are trained on the downstream tasks by fine-tuning all pre-trained parameters with minimal task-specific parameters (few parameters need to be learned from scratch). Two well-known and powerful fine-tuning PLMs are GPT-2 [17] and BERT [4].

BERT BERT [4] is a PLM based on a neural network architecture called Transformer [19]. BERT is trained on 0.8B words of the BooksCorpus and 2.5B words of English Wikipedia (3.3B words in total). BERT achieves SOTA on various downstream tasks, and we present the details of BERT in Appendix D.

[3] https://huggingface.co/bert-base-cased

GPT-2 Generative Pre-trained Transformer (OpenAI GPT) [16] is also based on the Transformer architecture. The model is trained on the BooksCorpus dataset containing more than 7,000 books from a variety of genres. GPT-2 [17] is an extension of the OpenAI GPT, in which the model size and data size are increased. As reported in [17], the numbers parameters in the models are: OpenAI GPT (117M), BERT (345M), and GPT-2 (1.5B). GPT-2 is trained on 40GB of text from the WebText (scraped from 45 million web pages). OpenAI GPT and GPT-2 models achieve SOTA on various NLP tasks such as language modeling, language inference, question answering, text classification, etc.

E.2 Deep generative models (DGMs)

Data can be in various kinds such as images, videos, texts, etc. A goal of building machine learning systems is to discover patterns and extract knowledge from data, then perform reasoning based on the observed data. One strategy is to approximate data distributions, which summarize all the information about the data in a finite set of parameters.[4]

The basic idea of generative modeling is to train a model, which can capture the underlying distribution of the data. Generative models provide a powerful mechanism for learning data distributions and simulating samples. Probabilistic generative model enables rich data to be explained in terms of simpler latent structure. The discovered structure can be helpful such as for the purposes of explanation, visualization, or improving generalization to unseen data. [22]

Deep learning approaches for generative models such as Generative Adversarial Networks (GANs) [5] and Variational Autoencoders (VAEs) [8] have shown their ability to learn smooth representations of images, text, audio etc, which can then be used to generate new and plausible data [18]. Generative models have many applications such as synthesizing images, videos, audios; text translation and summarization, drug synthesis, etc [2]. Besides the short-term applications, generative models hold the potential to automatically learn the natural features of a dataset.[5]

E.3 Variational autoencoders (VAEs)

VAE VAE is a powerful deep generative model to unsupervisedly learn a low-dimensional data (latent space) from a high-dimensional data [11], and has been applied in many downstream tasks such as classification, transfer learning, text generation, etc [3, 10]. VAE defines a joint distribution of observed inputs x and latent variables z with unknown prior distributions $p(z)$. Typically, the conventional and simple Gaussian prior can be chosen. The objective is to maximize the *Evidence Lower Bound* (ELBO):

$$\log\ p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log\ p_\theta(x|z)] - KL(q_\phi(z|x)||(p(z)), \tag{1}$$

where $q_\phi(z|x)$ is known as encoder (or variational posterior) which tries to encode the input x into a latent representation z; while $p_\theta(x|z)$ is known as decoder which tries to reconstruct the input x given the latent variable z.

The training objective is to minimize the reconstruction loss (compares the reconstructed output with the input x), and the regularization loss (KL divergence, which compares the learned posterior distribution (approximate variational posterior) in the latent space with the prior distribution, or in other words this regularisation term forces the learned posterior to be as close to the prior as possible).

E.4 VAE-based PLMs

OPTIMUS The OPTIMUS framework [10] is a large scale VAE-based language model. The goal of OPTIMUS is to learn a latent embedding space for sentence. OPTIMUS uses the pre-trained BERT model [4] to initialize the encoder's weights, and the pre-trained GPT-2 model [17] to initialize the decoder's weights. OPTIMUS is trained on 2M Wikipedia sentences. OPTIMUS has been shown to learn a more structured semantic space due to the use of the prior distribution in training. The pre-trained OPTIMUS model is fine-tuned in various downstream tasks and has shown the strengths in language understanding and language generation tasks.

BioVAE Our VAE-based pre-trained language model uses the OPTIMUS framework to train on a large amount of biomedical text with 34M sentences from 3.35M PubMed abstracts. The pre-trained model is fine-tuned and shows SOTA in biomedical text mining tasks as well enables to generate accurate biomedical texts.

[4] https://deepgenerativemodels.github.io/notes/introduction/

[5] https://openai.com/blog/generative-models/

F Training cost

In this work, we trained the OPTIMUS framework on a huge amount of data of 34M sentences. In terms of the model complexity, the OPTIMUS is a combination of the two large neural-based models (BERT with 340M parameters, and GPT-2 with 1.5B parameters), which result in a very large number of learning parameters.

For completing one BioVAE model that we released, we used 128 GPUs from the AI Bridging Cloud Infrastructure (ABCI).[6] In order to use these computing resources, it costs approximately 2000 ABCI points (or 3,600 USD). Therefore, training such large-scale models on a massive amount of text is costly. From using our BioVAE pre-trained models which are freely available, people can apply for a specific downstream task or their own tasks without such a large cost to train the large scale model from scratch.

References

[1] Iz Beltagy et al. SciBERT: A pretrained language model for scientific text. In *EMNLP-IJCNLP*, pages 3606–3611. ACL, November 2019.

[2] Sam Bond-Taylor et al. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *arXiv preprint arXiv:2103.04922*, 2021.

[3] Samuel Bowman et al. Generating sentences from a continuous space. In *CONLL*, pages 10–21, 2016.

[4] Jacob Devlin et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

[5] Ian Goodfellow et al. Generative adversarial nets. *NeurIPS*, 27, 2014.

[6] Yu Gu et al. Domain-specific language model pretraining for biomedical natural language processing. *arXiv preprint arXiv:2007.15779*, 2020.

[7] Andrej Karpathy et al. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015.

[8] Diederik P Kingma et al. Auto-encoding variational bayes. In *ICLR*, 2013.

[9] Jinhyuk Lee et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 09 2019. btz682.

[10] Chunyuan Li et al. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *EMNLP*, pages 4678–4699. ACL, 2020.

[11] Ruizhe Li et al. On the low-density latent regions of vae-based language models. In *NeurIPS 2020 Workshop on Preregistration in Machine Learning*, pages 343–357. PMLR, 2021.

[12] Tomas Mikolov et al. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119, 2013.

[13] Jeffrey Pennington et al. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[14] Matthew Peters et al. Deep contextualized word representations. In *NAACL*, pages 2227–2237, 2018.

[15] Xipeng Qiu et al. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26, 2020.

[16] Alec Radford et al. Improving language understanding by generative pre-training. 2018.

[17] Alec Radford et al. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

[18] Bidisha Samanta et al. Nevae: A deep generative model for molecular graphs. *JMLR*, 2020.

[19] Ashish Vaswani et al. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[20] Alex Wang et al. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.

[6] https://abci.ai/

[21] Ye Zhang et al. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

[22] James Yang Zou et al. Priors for diversity in generative latent variable models. 2012.