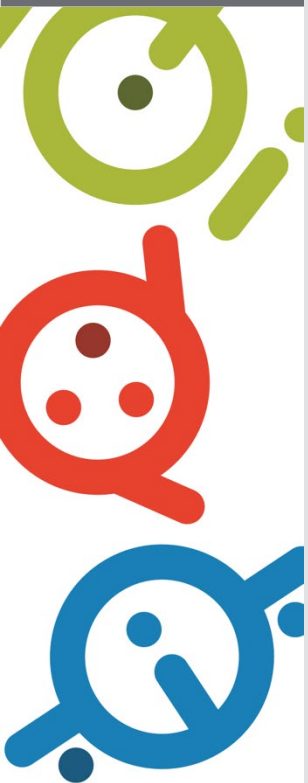
Carnegie Mellon University
Software Engineering Institute



Research Review 2021

README

A Learned Approach to Augmenting Software Documentation

Thursday, 30 September 2021

Dan DeCapria, PI
Senior Data Scientist

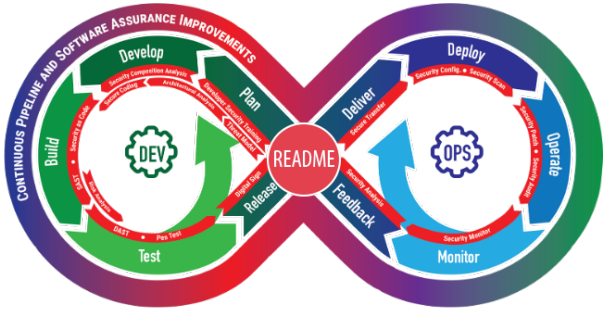
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM21-0854

DoD Impact: Documentation with DevSecOps

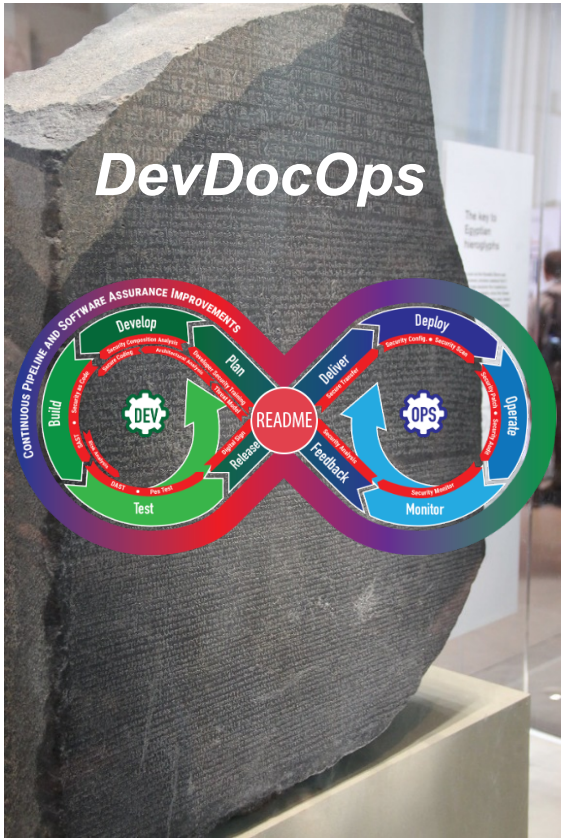


DevSecOps software documentation processes are inadequate, costly in time, and difficult to verify quantitatively.

An estimated 10%, minimum, of software systems' overall costs are allocated to software documentation tasking internally through DoD contractors. This estimate can be as high as 25% of overall costs when considering cybersecurity and DoD security requirements, if tracked at all.

The README proof of concept (POC) is a strategic step toward a generative software documentation process in the modern DoD DevSecOps SDLCs.

Problem Description

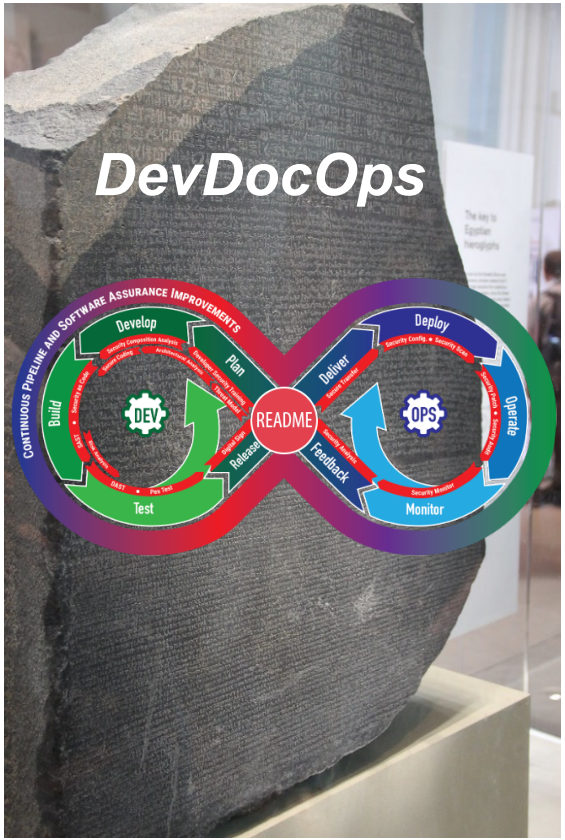


Research a machine learning (ML) application to generate the descriptive content for automated software documentation.

The need for an ML appliance for generating software documentation from source code for viability in modern DevSecOps SDLCs

- Translation from source code to language
- Model reuse, modularity
- Prior-art considerations, CV domains

Problem Description



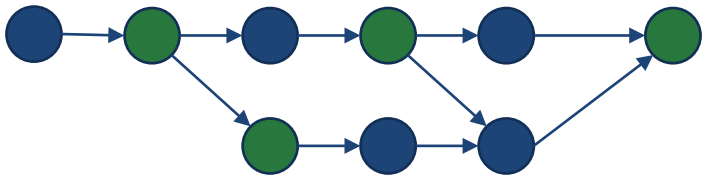
Research a machine learning (ML) application to generate the descriptive content for automated software documentation.

Introduce the Matryoshka Technique:

A modular approach, using pretrained models, with a nested model for learning a shared embedding suitable for cross-domain latent translation between source code and natural language descriptors

README: Matryoshka Technique

README



GIT Repository Language Co-Occurrence

Martyoshka Technique

Variational Auto-Encoder (VAE)
- Conditional VAE (CVAE)

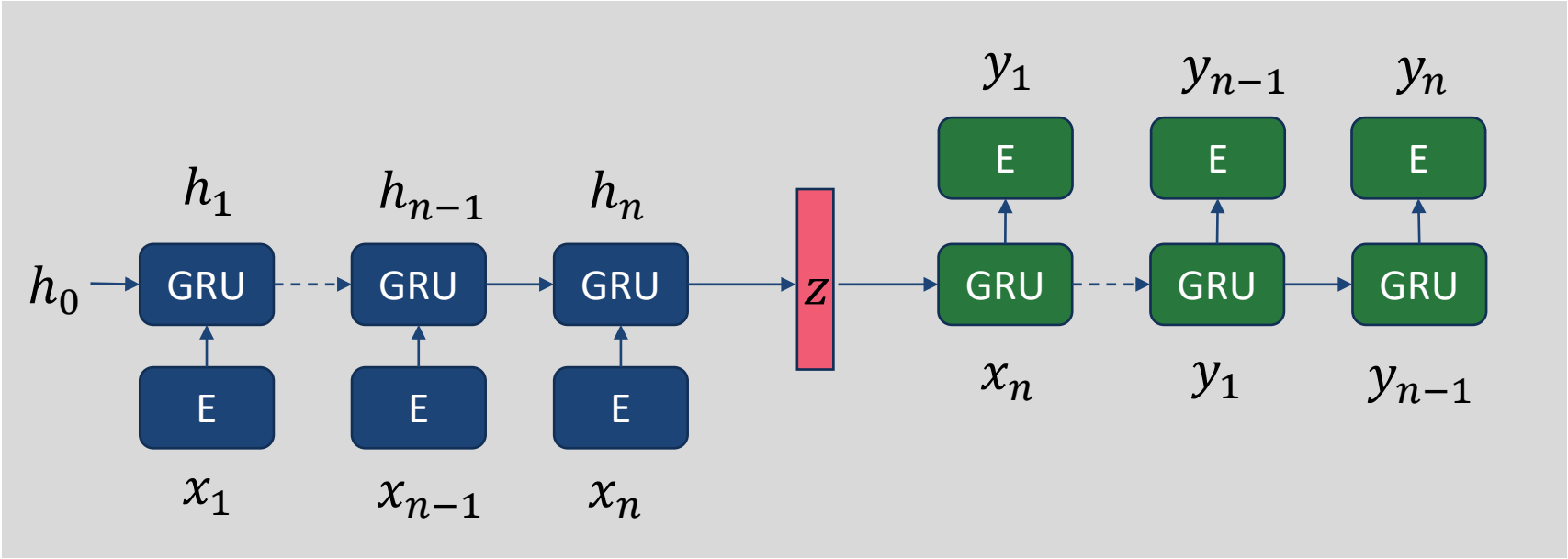
POC cross-domain translation
- Prior art in CV domain
- Source code (Python 3.8 CFG)
- Natural language descriptors

Open-source GitHub repositories
- GIT branch commit histories

Existing software engineering lexicons
- StackOverflow word embedding

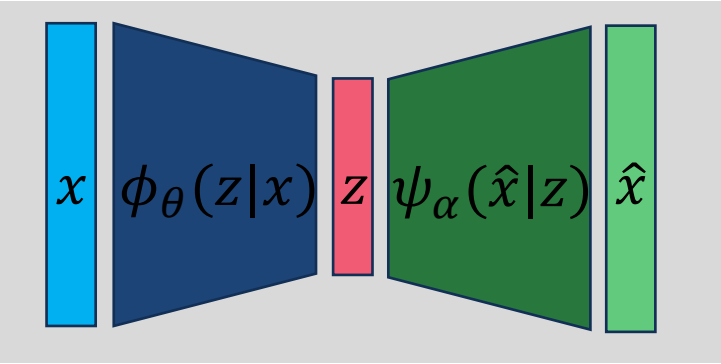
Prototype DevSecOps MVP Service
- RESTful endpoint exemplar

Notional Architectures

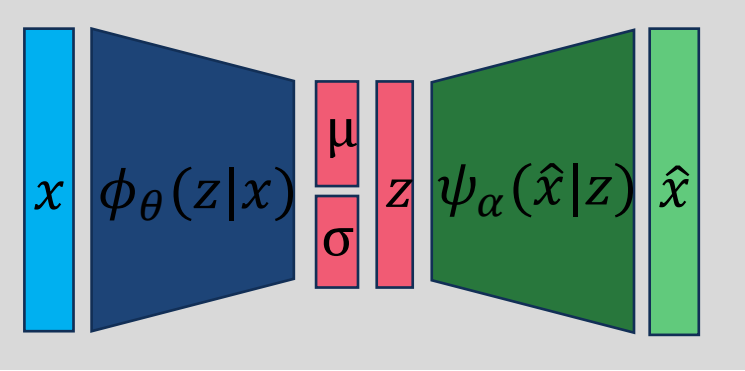


Gated Recurrent Unit Network

Notional Architectures



Auto-Encoder

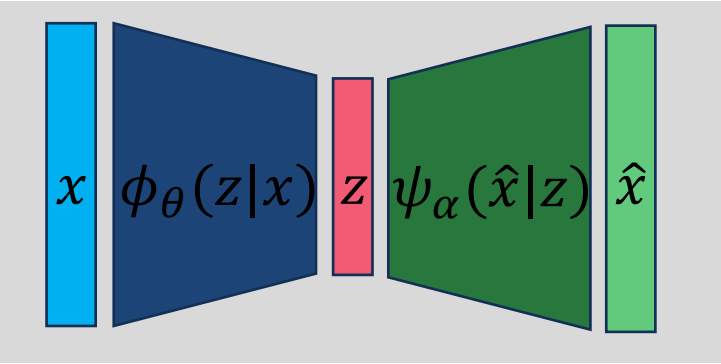


Variational Auto-Encoder

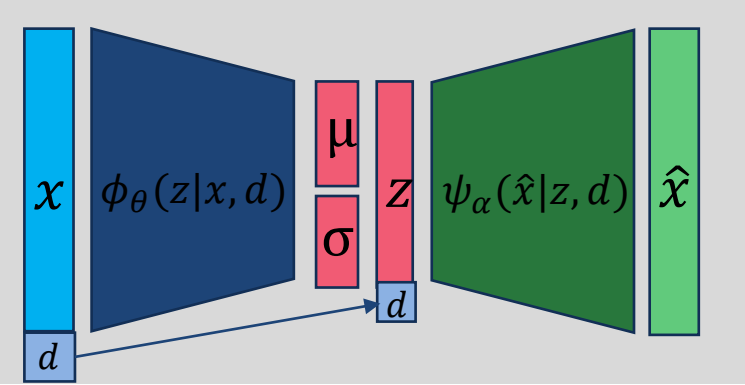
$$z = \mu + \sigma \cdot \epsilon$$

$$\epsilon \sim N(0,1)$$

Notional Architectures



Auto-Encoder



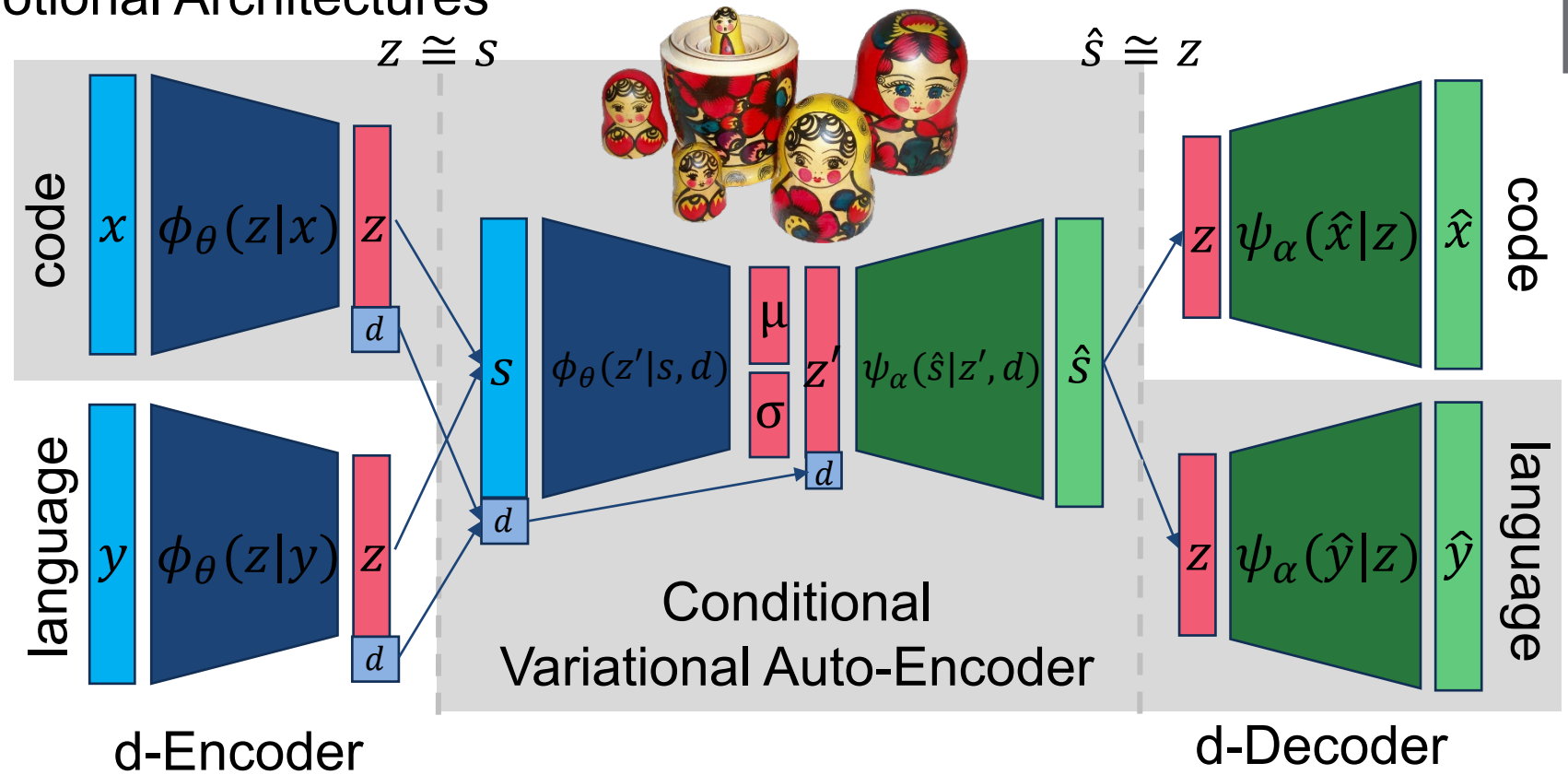
Conditional
Variational Auto-Encoder

$$z_d = \mu_d + \sigma_d \cdot \epsilon$$

$$\epsilon \sim N(0,1)$$

$$d \in \{0,1\}$$

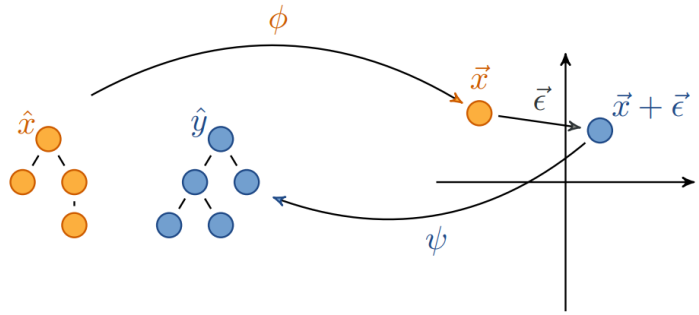
Notional Architectures



Martyoshka Technique: Pretrained Model Reuse

Tian, Y., & Engel, J. Latent Translation: Crossing Modalities by Bridging Generative Models. *arXiv*:1902.08261. 2019.

Pretrained Models: AST2VEC COTS



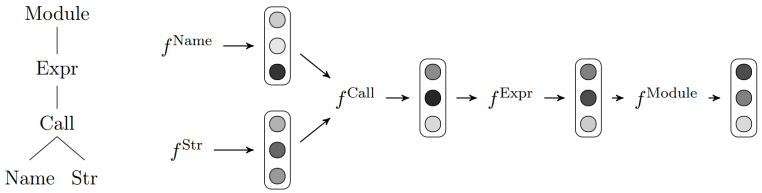
AST Reconstruction

Dataset

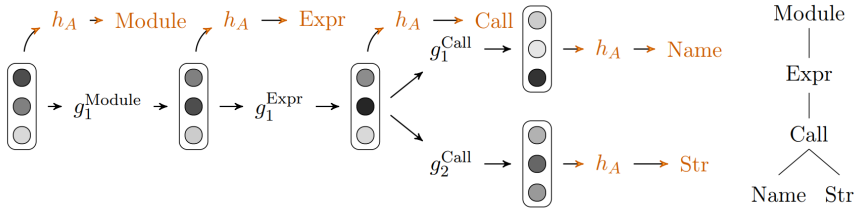
- Python Programs
  - National Computer Science School [448k]

Optimization: Minimize

$$-\alpha * \log\left[p_\psi(\hat{x}|\phi(\hat{x}+\epsilon))\right]$$
$$+\beta * \Delta\left[KL(\phi(\hat{x})+\epsilon)\right]$$



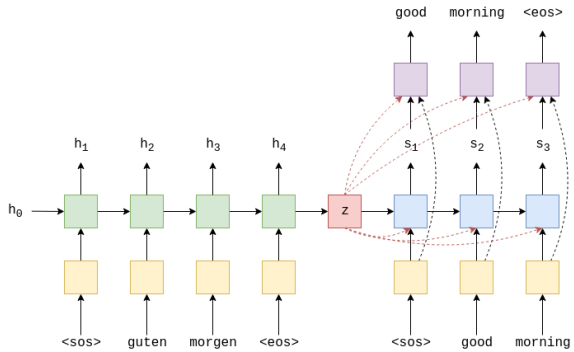
Python CFG GRU Encoder

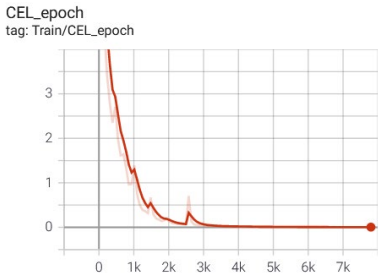


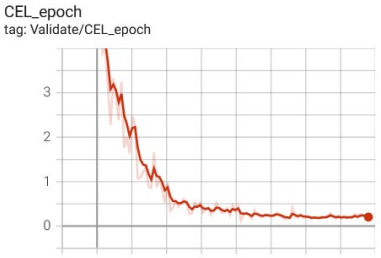
Python CFG GRU Decoder

Paaßen, B.; McBroom, J.; Jeffries, B.; Koprinska, I.; & Yacef, K. Mapping Python Programs to Vectors using Recursive Neural Encodings. *Journal of Educational Datamining*. 2021. In press. https://gitlab.com/bpaassen/ast2vec

Pretrained Models: Seq2Seq SO T&V







Seq2Seq SO Reconstruction

Dataset

- StackOverflow
  - Posts Archive [53M, ~85GB]
  - Gensim word2vec E[1.7M x 200]
  - Pareto Popular e[100k x 200, ~115MB]
  - SpaCy Modified ENG-Tokenizer Ruleset

Optimization: Minimize over Vocabulary

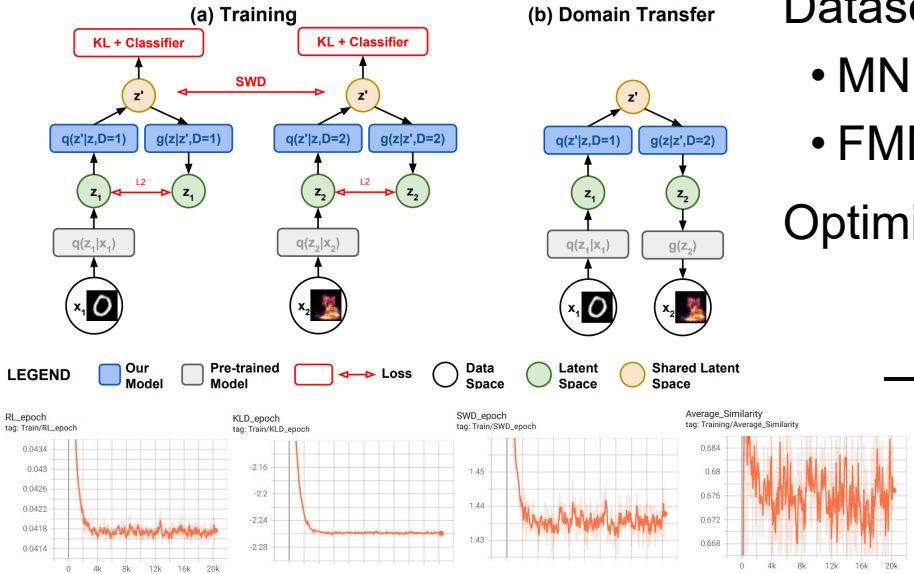
$$CEL(\hat{e}, v)$$

Cho, K.; Merrienboer, B.V.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; & Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *EMNLP*. 2014.

Efstathiou, Vasiliki; Chatzilenas, Christos; & Spinellis, Diomidis. Pages 38–41. Word embeddings for the software engineering domain. In *Proceedings of the 15th International Conference on Mining Software Repositories (MSR '18)*. 2018. DOI: https://doi.org/10.1145/3196398.3196448

[terminal, directive, option, python, errno, operation, antialias, collector, …] -> [828, 2437, 184, 4, 576, 213, 3310, 2020, …]

Nested Model: CVAE Image T&V



CVAE Reconstruction

Datasets

- MNIST [(1,28,28), 60k/10k]
- FMNIST [(1,28,28), 60k/10k]

Optimization: Minimize ELBO, SWD, ~~CL~~*SL*

$$\alpha * MSE(\hat{s}_d, s_d)$$

$$-\beta * \Delta[KL(1 + \sigma_d - \mu_d^2 - e^{\sigma_d})]$$

$$\gamma * W_2^2(\pi(z'_d, \omega), \pi(z'_{\bar{d}}, \omega))$$

$$\delta * cosine\_sim(s_d, \hat{s}_d)$$

Tian, Y., & Engel, J. Latent Translation: Crossing Modalities by Bridging Generative Models. arXiv:1902.08261. 2019.

Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*. Volume 29. Number 6. Pages 141–142. 2012.

Xiao, Han et al. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *ArXiv* abs/1708.07747. 2017.

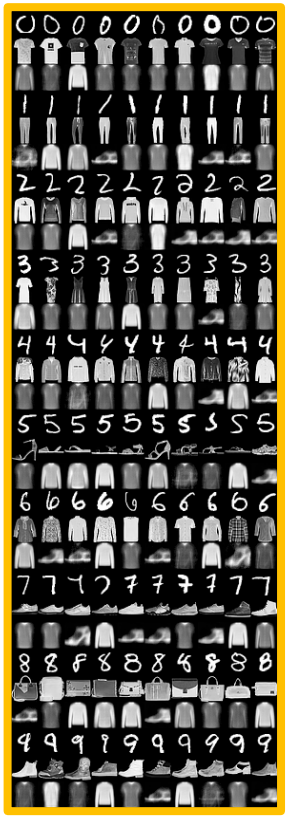
Nested Model: VAE versus CVAE Image T&E

MNIST Source Images

FMNIST Source Images

Pretrained Latent Translation VAE: Source, Target, Reconstruction

Pretrained Latent Translation CVAE: Source, Target, Reconstruction

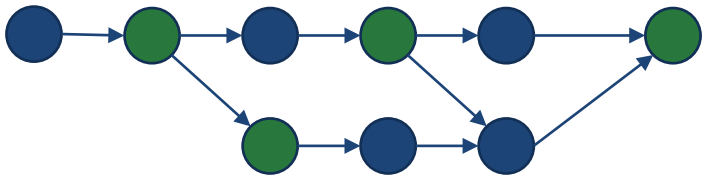


MNIST Reconstructed Images

FMNIST Reconstructed Images

README: Matryoshka Technique

README

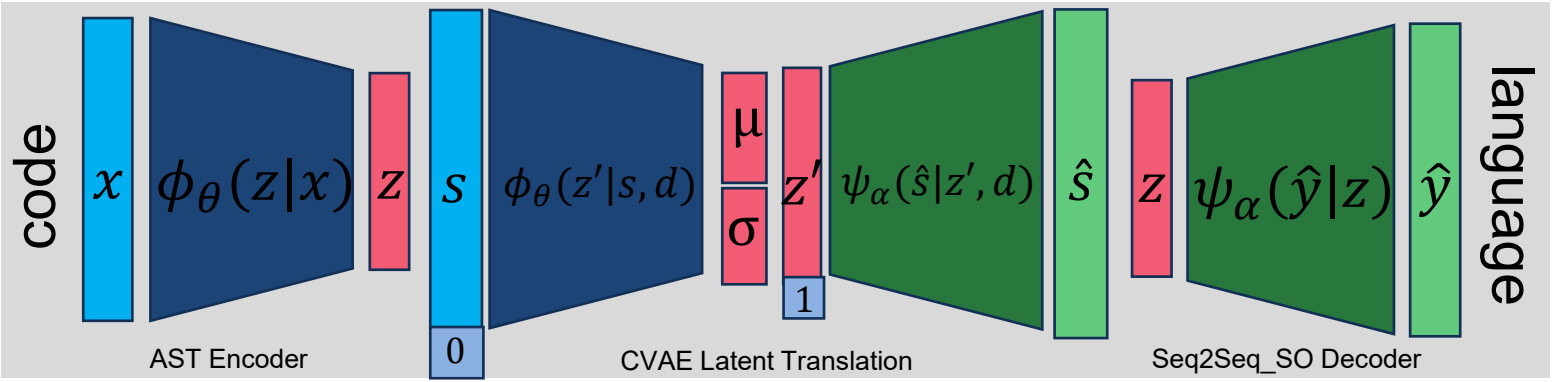


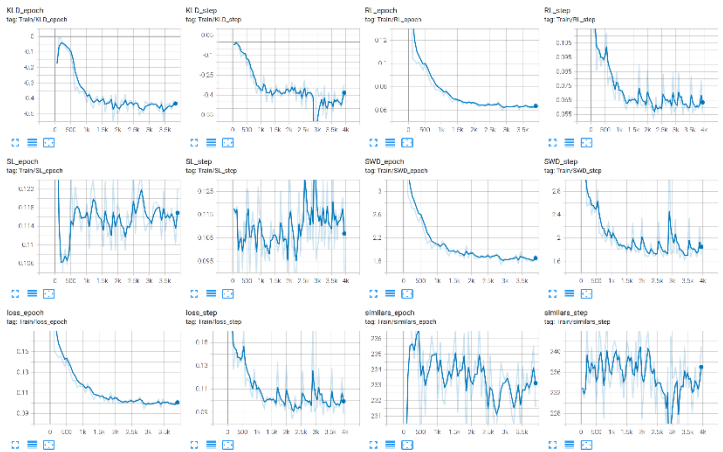
GIT Repository Language Co-occurrence

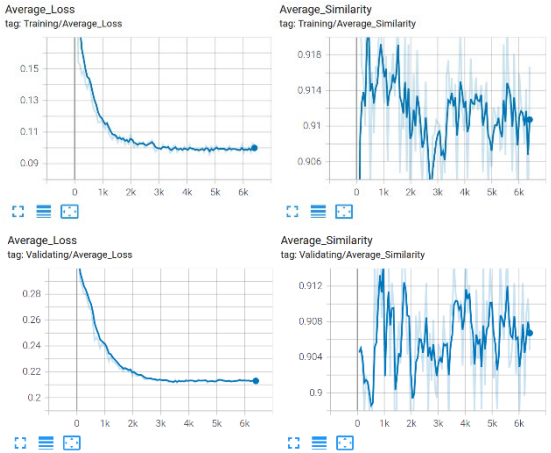
Martyoshka Technique

- **DataOps**
- **MLOps**
- **COTS Pretrained Models**
  - AST2VEC T&E
  - Seq2Seq_SO T&V, T&E
- **Latent Translation Paradigm**
  - CVAE T&V, T&E
- **README POC v0.3**
  - Datum (AST, [vocab])
  - Learn z' Shared Embedding T&V
- **README DevSecOps Prototype**

README: Model POC T&V





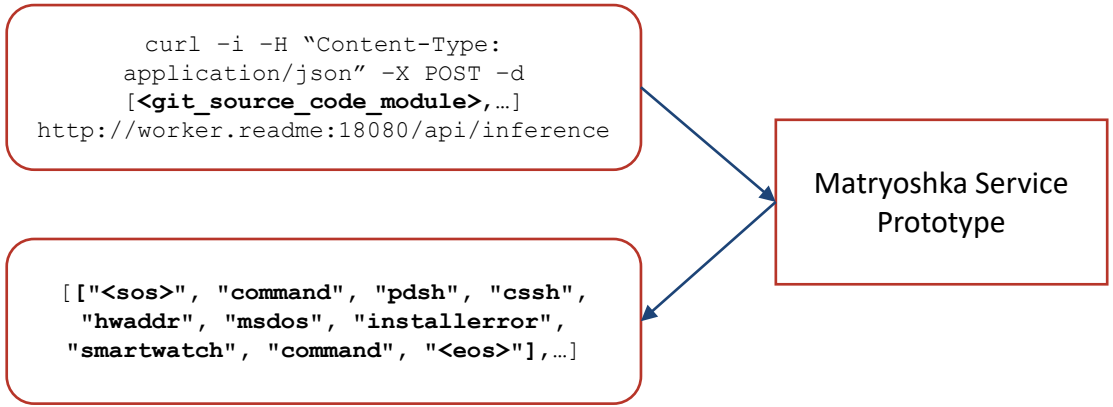


~90% similarity in 70% co-domain reconstruction coverage

README: DevSecOps MVP Prototype

README Matryoshka MVP DevSecOps Prototype

- Containerized README POC v0.3 SaaS
- Usage: POST JSON Array source code module(s) payload
- Response: JSON Array of translated SWE lexicon token(s)
- Additional examples in source code Python Notebooks and READMEs



Conclusions and Future Considerations

DevSecOps and Documentation Process Prototype Service

- README: A Learned Approach to Augmenting Software Documentation, preprint

Martyoshka Technique: POC Demonstration

- Encoder: AST2VEC Python 3.8 CFG
- Decoder: Seq2Seq_SO StackOverflow Lexicon
- CVAE Shared Latent Embedding Model
- README T&V Supporting Operations
- DevSecOps Prototype Service Exemplar

DevSecOps and DevDocOps CI/CD Service Prototype

Additional Modalities and Use Cases

Software Factories and Industry Outreach and Support

SEI Team

Carnegie Mellon University Software Engineering Institute

Dan DeCapria
Senior Data Scientist
SEI AI Division

Tina Sciullo-Schade
Research Project Manager
SEI AI Division

Tanisha Valerie Smith
Technical Manager
SEI AI Division

Hasan Yasar
Technical Director
SEI Software Solutions Division

Ipek Ozkaya
Technical Director
SEI Software Solutions Division

Violet Turri
Assistant Software Developer
SEI AI Division

Will Nichols
Infrastructure Engineer
SEI AI Division

Alex Van Deusen
Assistant Design Researcher
SEI AI Division

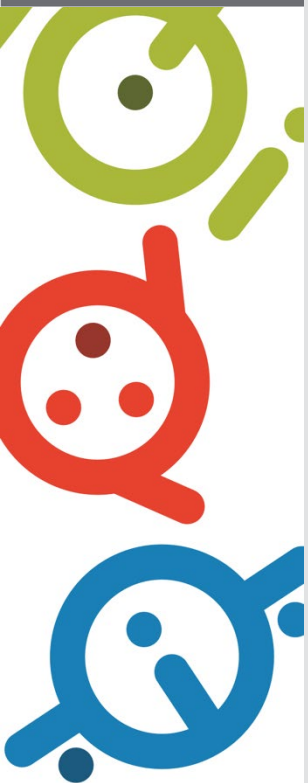
Andrew Mellinger
Senior Software Developer
SEI AI Division

Jay Palat
Senior Engineer
SEI AI Division

Contact Information

Dan DeCapria, PI
Artificial Intelligence Division
Software Engineering Institute
Carnegie Mellon University
info@sei.cmu.edu

Carnegie Mellon University
Software Engineering Institute



Research Review 2021

Backup Slides