Binary alias analysis using transformers

Laureline Dubucq
Kunj Haria

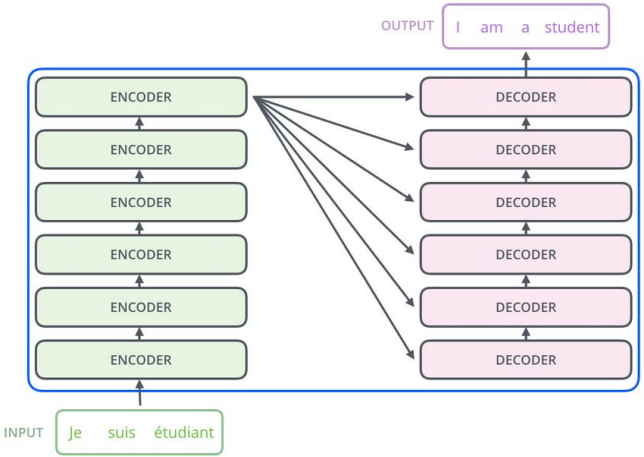
Contents

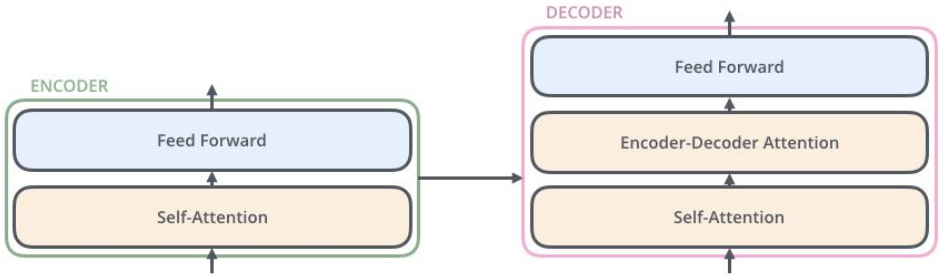
Introduction

- **Binary analysis:**
  - Type of machine code review
  - Assess content and structure
  - No need of source code
- **Alias analysis:**
  - used to determine if a storage location may be accessed in more than one way.
  - Eg. Two pointers are said to be aliased if they point to the same location.
- **Postmortem analysis:**
  - Done after a program crash
  - Problem: incomplete data flow

Introduction - *cont'd*

- **Transformers:**
  - Seq2seq deep learning model
  - Handle long-term dependencies using self-attention layers
  - Can be parallelized





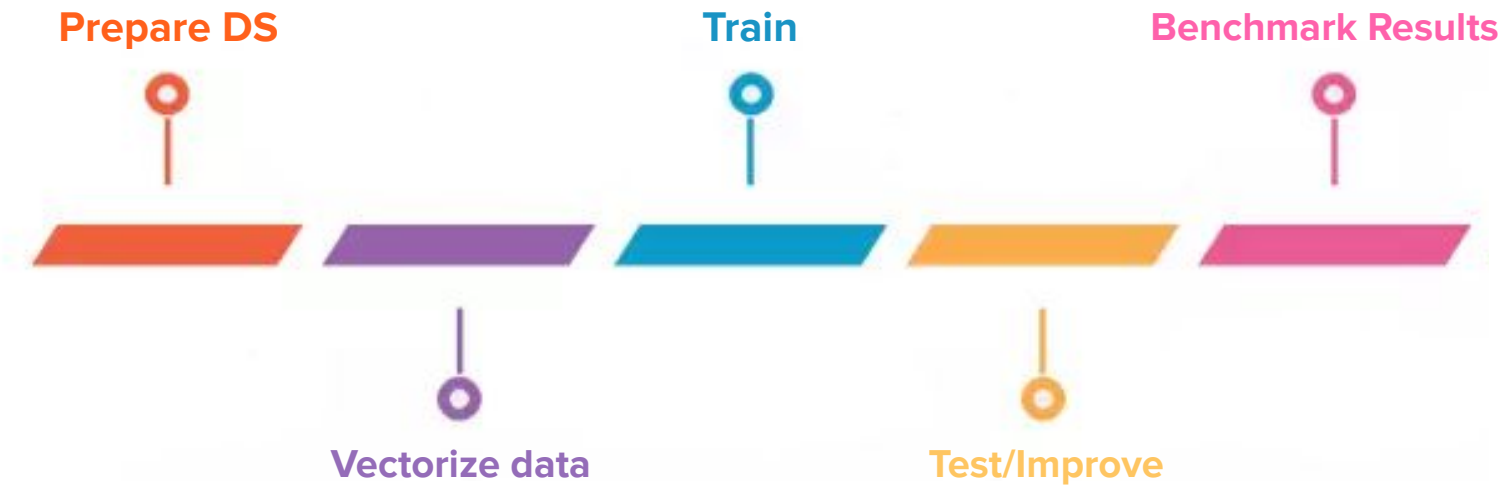
➜ http://jalammar.github.io/illustrated-transformer/

Motivation

- VSA: not efficient for postmortem analysis
- DeepVSA introduced to fix that problem
  - Uses bidirectional LSTMs to learn which memory region corresponds to each instruction
- Compressive Transformers outperformed previous seq2seq models in many tasks

➜ **Goal: improve DeepVSA using Transformers**

Process


Prepare DS
Train
Benchmark Results
Vectorize data
Test/Improve

Questions ??

Bibliography

Transformers:

- How transformers work
- Compressive transformer vs LSTM
- Attention is all you need ➔ first paper presenting transformers - 2017
- Comprehensive guide to Transformers

DeepVSA:

- DEEPVSA: Facilitating Value-set Analysis with Deep Learning for Postmortem Program Analysis
- Presentation
- Github Project