A Pseudo-Labelling Approach for Unsupervised Domain Adaptation on Assembly Code

Nirav Diwan

High Level Language

Assembly Code

```
0300 0000 fa74 0300 5001 0000 cb1e 0000
fa07 0000 1c00 0000 8000 0000 0000 0000
0458 0300 0800 0000 0410 0000 6462 0300
fcff 0300 6807 0000 e35a 7b01 ff03 0000
0100 0000 ec74 0300 0000 0000 0000 0000
0dc0 a0e1 0058 2de9 0cb0 a0e1 ff5f 2de9
f08f 9fe5 0000 c8e5 0010 a0e3 e08f 9fe5
0010 c8e5 0010 a0e3 2310 c8e5 0010 a0e3
2410 c8e5 0010 a0e3 2510 c8e5 0010 a0e3
2610 c8e5 0010 a0e3 b712 c8e1 0010 a0e3
2910 c8e5 ac8f 9fe5 0010 d8e5 0400 2de5
```

Binary Code

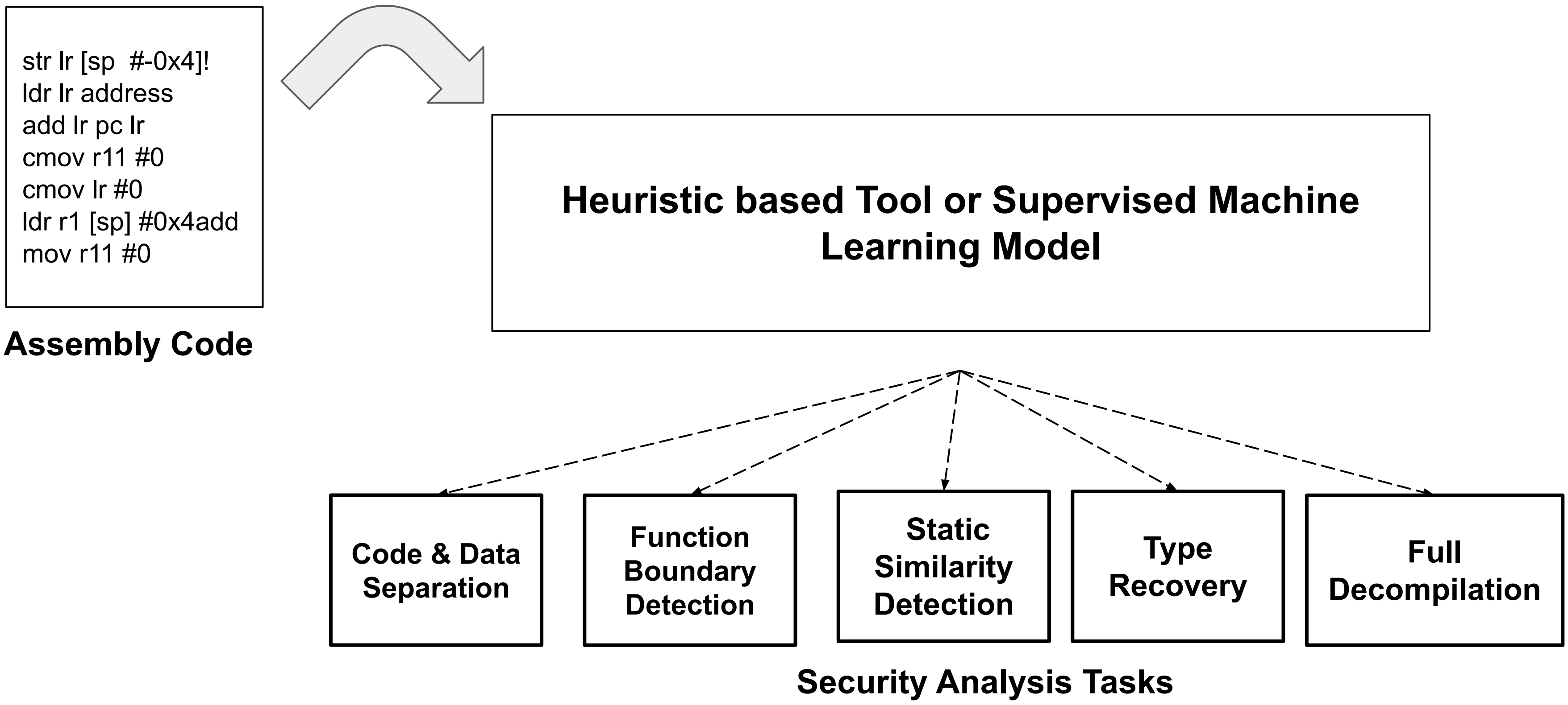
Format of Assembly Code

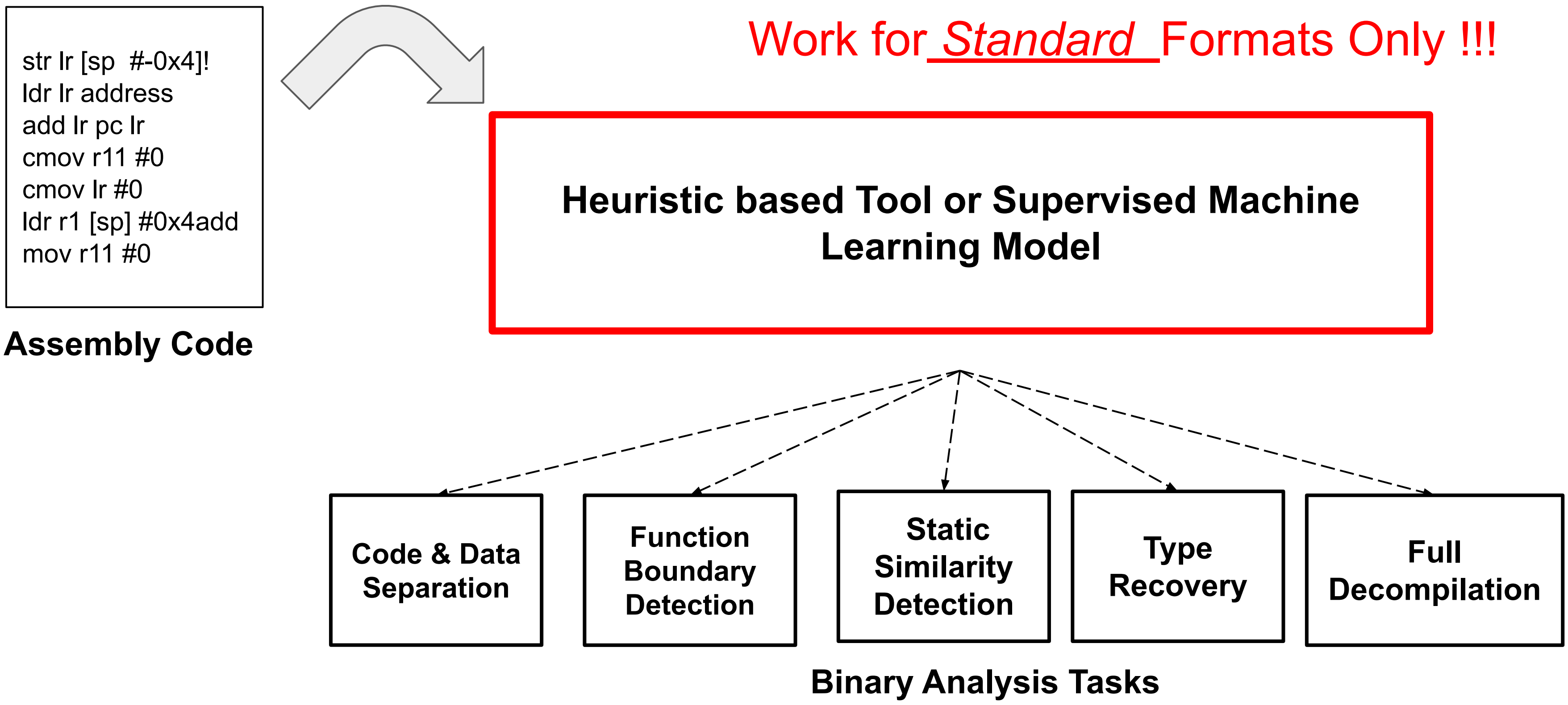
```
cmov r11 #0
add r1 r2
sub r11 #0x4
moveq r11 #0x4
```

Opcode

Parameters

Flags


str lr [sp #-0x4]!
ldr lr address
add lr pc lr
cmov r11 #0
cmov lr #0
ldr r1 [sp] #0x4add
mov r11 #0
Assembly Code
Heuristic based Tool or Supervised Machine Learning Model
Code & Data Separation
Function Boundary Detection
Static Similarity Detection
Type Recovery
Full Decompilation
Security Analysis Tasks



str lr [sp #-0x4]!
ldr lr address
add lr pc lr
cmov r11 #0
cmov lr #0
ldr r1 [sp] #0x4add
mov r11 #0
Assembly Code
Work for Standard Formats Only !!!
Heuristic based Tool or Supervised Machine Learning Model
Code & Data Separation
Function Boundary Detection
Static Similarity Detection
Type Recovery
Full Decompilation
Binary Analysis Tasks

Standard | Non - Standard

Blowing up !!!

Standard

Most softwares based on common Programming Languages that run on popular OS

Non-Standard

Custom device software -

- Modern Day Router, Bluetooth headphones.
- IoT based device software - TV, Car, Ovem
- Large Ecosystems - Power Grids, Dams

Benkraouda, Hadjer, et al. "BinSeg: Leveraging Semantic Segmentation for Code and Data Separation in Non-standard Binary Formats."

Goal : Domain Adaptation for Non-Standard Assembly

Challenges

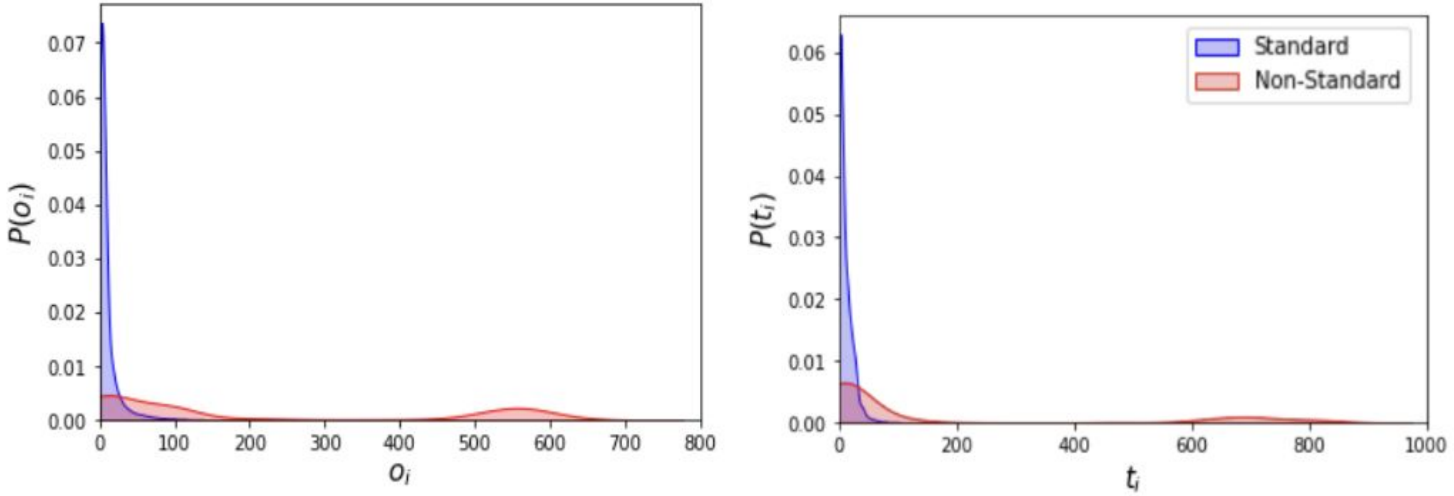
1. Low Resource
2. Expensive Manual Cost
3. Out of Vocabulary
4. Long Range Dependence

Challenges

1. Low Resource
2. Expensive Manual Cost
3. Out of Vocabulary
4. Long Range Dependence

Out of Vocabulary Words (OOV)

Unigram Probability



(a) Opcode,
$d_{TV} = 0.0676$
$d_{JS} = 0.0619$

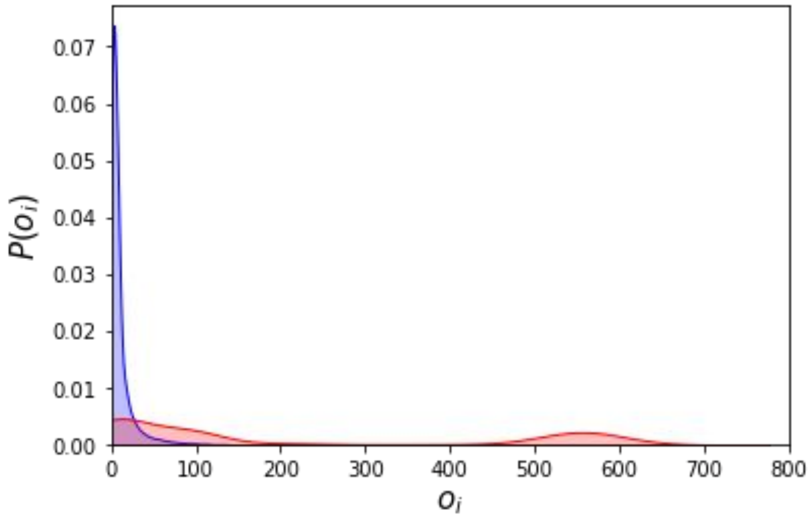
(b) All Tokens
$d_{TV} = 0.0448$
$d_{JS} = 0.0459$

* X - axis is sorted according to the descending order of standard vocabulary size

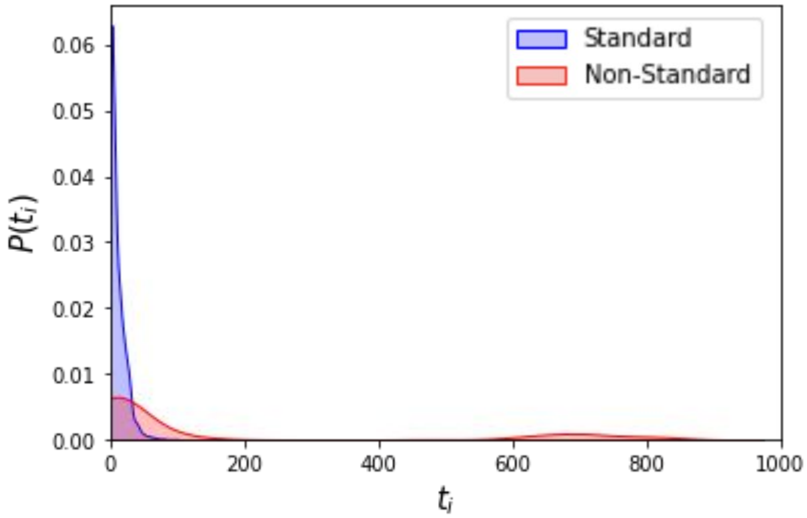
Verdú, Sergio. "Total variation distance and the distribution of relative information." *2014 Information Theory and Applications Workshop (ITA)*. IEEE, 2014.

Out of Vocabulary Words (OOV)

Unigram Probability



(a) Opcodes



(b) All tokens

$$d_{TV} = 0.676$$
$$d_{JS} = 0.619$$

$$d_{TV} = 0.448$$
$$d_{JS} = 0.459$$

* X - axis is sorted according to the descending order of standard vocabulary size

Verdú, Sergio. "Total variation distance and the distribution of relative information." *2014 Information Theory and Applications Workshop (ITA)*. IEEE, 2014.

Challenges

1. Low Resource
2. Expensive Manual Cost
3. Out of Vocabulary
4. Long Range Dependence

Analogous to Domain Adaptation for Low Resource Languages

Standard → **Non - Standard** ~ **High Resource** → **Low Resource**

1. Low Resource ✓
2. Expensive Manual Cost ✓
3. Out of Vocabulary ✓
4. Long Range Dependence ✓

Methodology

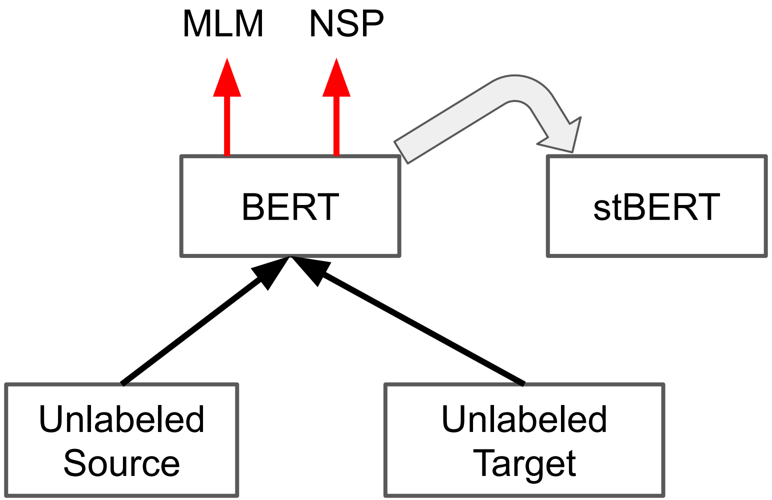
Similar to Unsupervised Domain Adaptation for Low Resource Languages

Task : Code & Data Separation

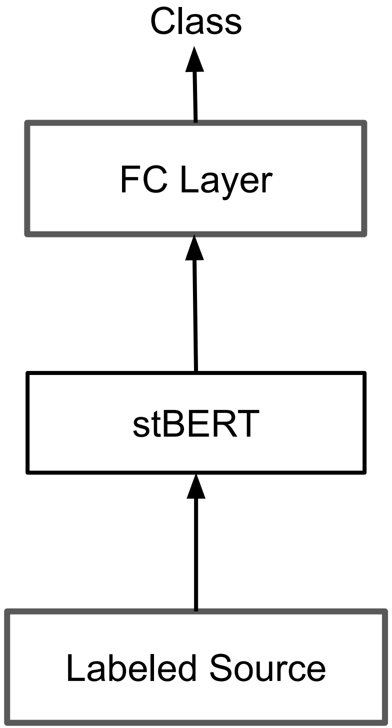
- Given an instruction - Is it a code or data instruction?
- Balanced Binary Classification Task
- Source: Standard
    - Labeled : 10k samples
        - 5K code
        - 5K data
    - Unlabeled: 20M samples
- Target: Non-Standard
    - Labeled: 2k samples
        - 1k code
        - 1k data
    - Unlabeled: 1M samples

Benkraouda, Hadjer, et al. "BinSeg: Leveraging Semantic Segmentation for Code and Data Separation in Non-standard Binary Formats."

Common Approach - stBERT + FT



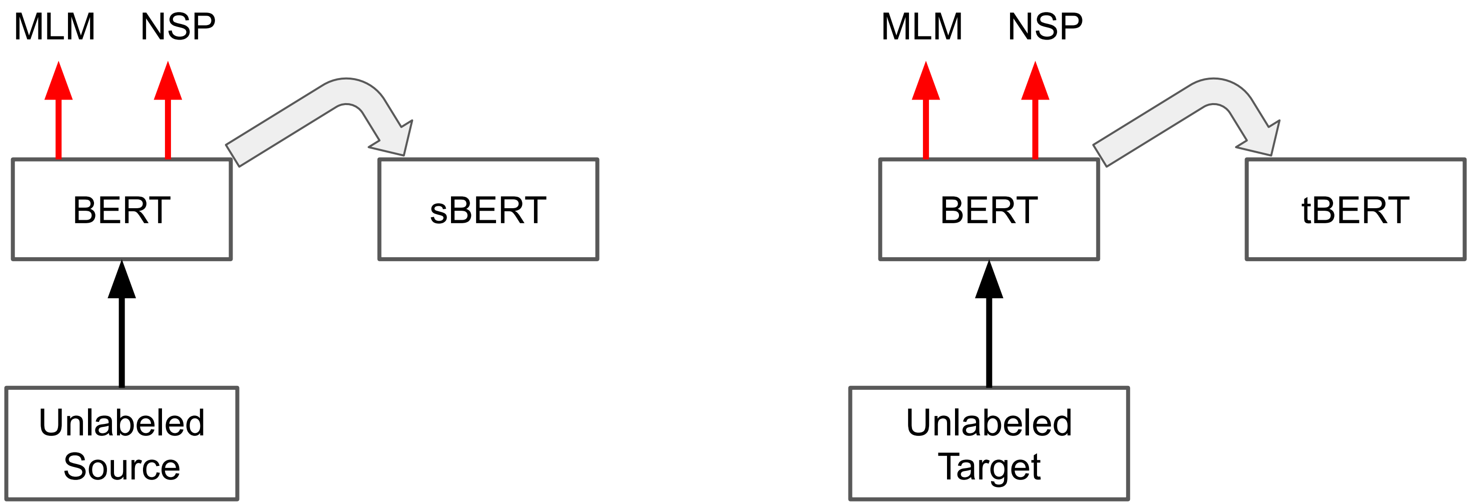
(a) Pre-train combined source-target BERT → stBERT



(b) Finetune st-BERT for Task

Model	**Domain**	**F1 (Val)**	**F1 (Test)**
stBERT - FT	S	0.99	0.96
	T	0.72	0.70

Joint Fine-tuning (JFT) - Pseudo Labelling Approach (PsL)

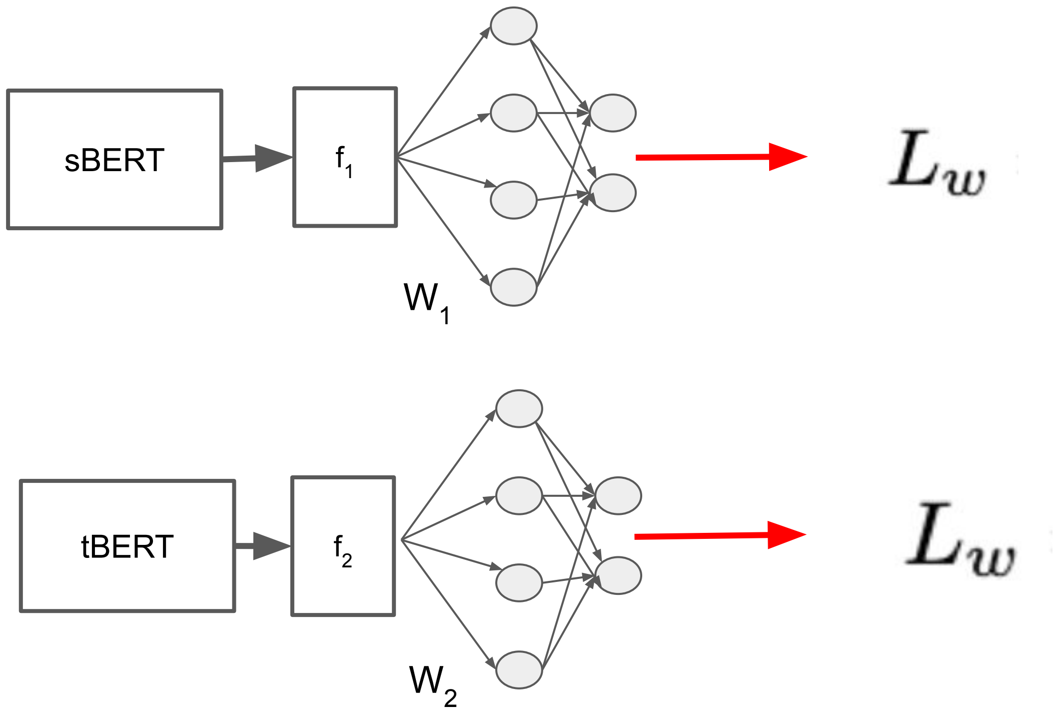


(a.1) Pre-train source BERT → sBERT

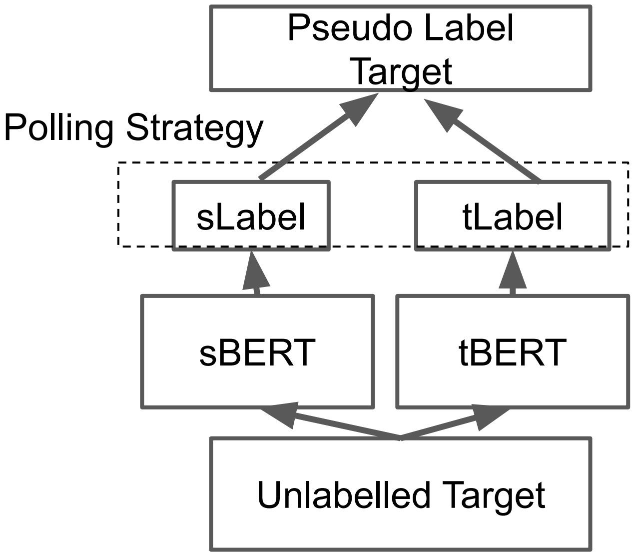
(a.2) Pre-train source BERT → tBERT

Tianyu Chen, Shaohan Huang, Furu Wei, and Jianxin Li. 2021. Pseudo-Label Guided Unsupervised Domain Adaptation of Contextual Embeddings. In Proceedings of the Second Workshop on Domain Adaptation for NLP, pages 9–15, Kyiv, Ukraine. Association for Computational Linguistics.

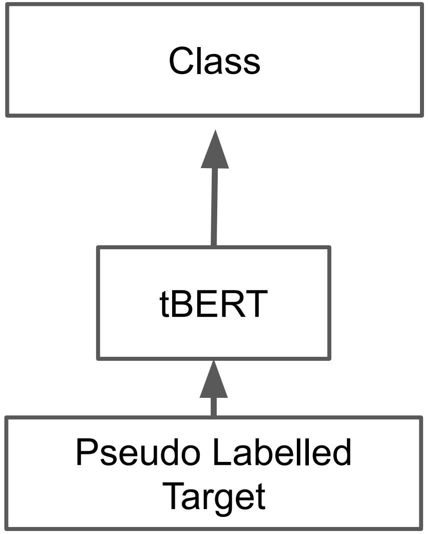
JFT + PsL



JFT + PsL

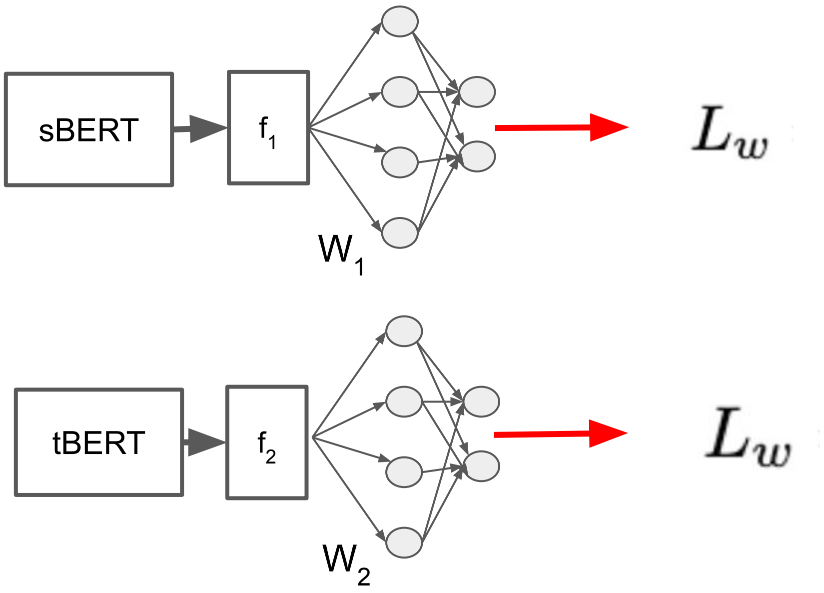


(c) Creation of Pseudo Labeled Target Dataset



(d) Fine-tuning of Pseudo Labeled Target Data on tBERT

JFT - PsL Loss Function



$$L_w = \boxed{CE_{loss}\left(s\left(x_i\right), y_i\right)} + \boxed{CE_{loss}\left(ns\left(x_i\right), y_i\right)} + \boxed{\lambda \left|W_1^T W_2\right|}$$

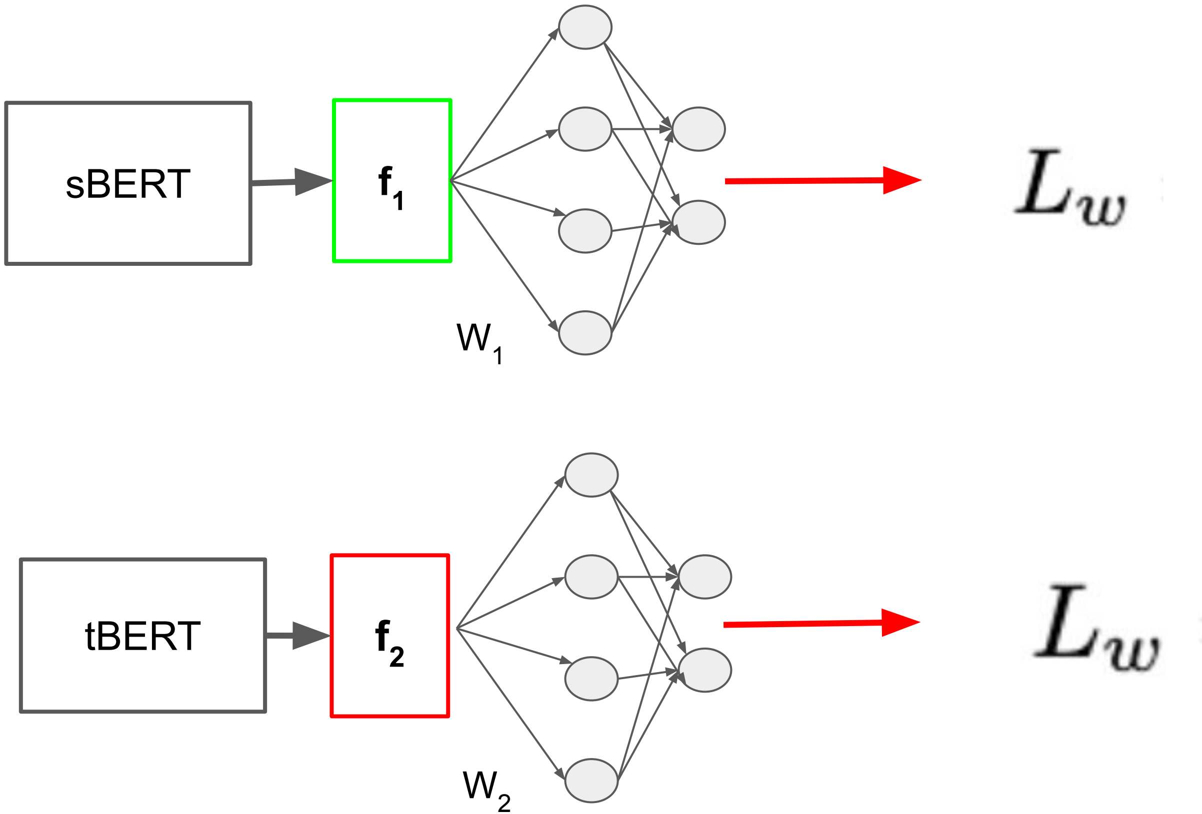
Source model loss on Source Data

Target model loss on Source Data

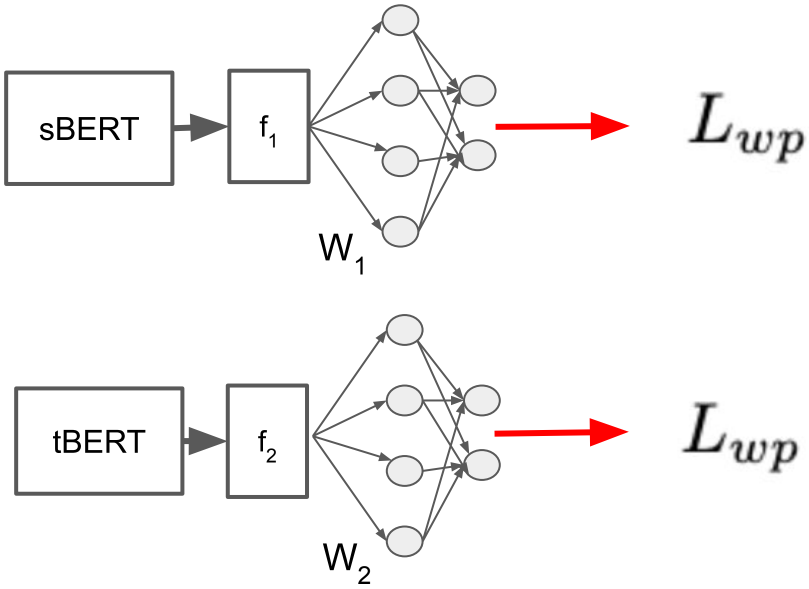
Learn "Different Features"

Model	Domain	F1 (Val)	F1 (Test)
stBERT - FT	S	0.99	0.96
	T	0.72	0.70
JFT - PsL	S	0.99	0.98
	T	0.75	0.73

Distance between "Initial Feature Space"



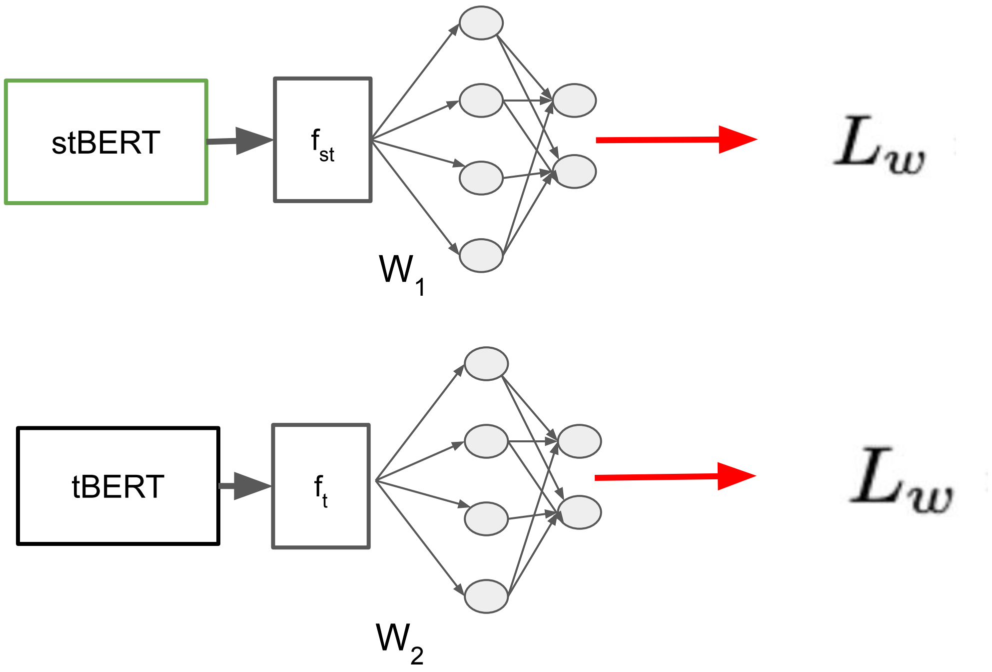
JFT - PsL - L_p



$$L_{wp} = \boxed{CE_{loss}\left(s\left(x_i\right), y_i\right)} + \boxed{CE_{loss}\left(ns\left(x_i\right), y_i\right)} + \alpha \cdot \boxed{\left|W_1^T W_2\right|} + \beta \cdot \boxed{\| \ f_1 - f_2 \ \|_p}$$

from same “initial”
Feature Spaces

JFT - PsL - stBert



Saito, Kuniaki, Yoshitaka Ushiku, and Tatsuya Harada. "Asymmetric tri-training for unsupervised domain adaptation." *International Conference on Machine Learning*. PMLR, 2017.

Results

Future Work

- Study the representations of source and target domains in more depth
- Try Pivot Based Domain Adaptation, Domain Invariant Representations

Also in Report

- Literature Review
- Optimization on Tokenization