Domain Adaptation Using Transfer Learning Techniques

Ashish Gupta
University of California, San Diego
asg007@ucsd.edu

Introduction to Scrabble

- Building Management System (BMS)
- Every BMS point is associated with a unique source identifier (srcid)

SourceIdentifier	VendorGivenName	BACNetName	BACNetUnit
123-456	RM-101.ZNT	VAV101 ZoneTemp	64

- Brick is a metadata schema which provides for a complete vocabularies and relationships

```
{
 ("ex:znt", "rdf:type", "brick:Zone_Temperature_Sensor"): 0.9,
 ..
}
```

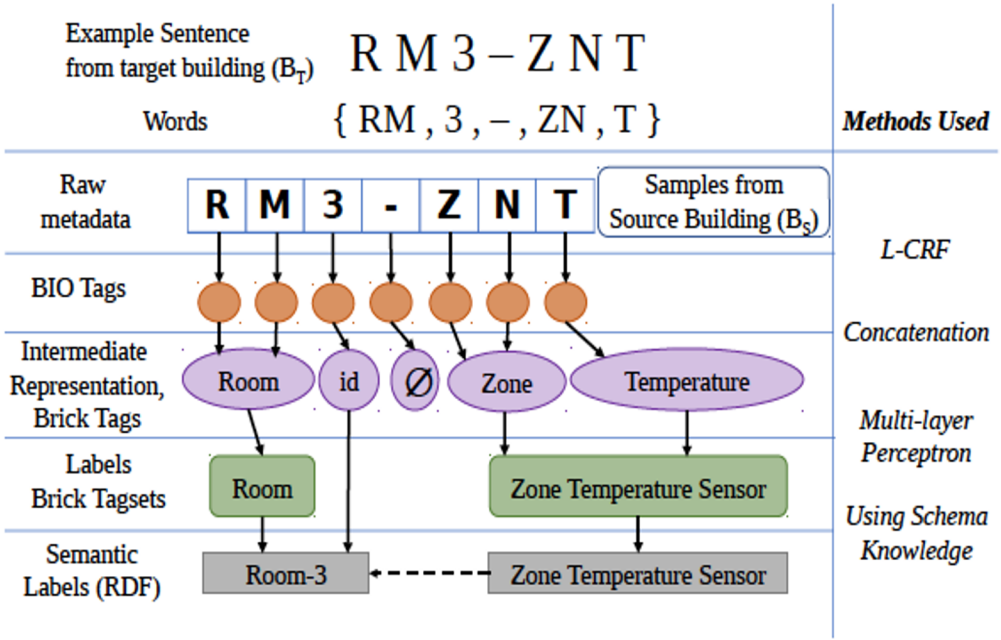
- Scrabble framework forms an interface between these two

Introduction to Domain Adaptation

- Sub-Discipline of Machine learning
- Classifier trained only on a single domain will suffer degradation on the second domain
- Domain Adaptation eliminates the difference in these distributions

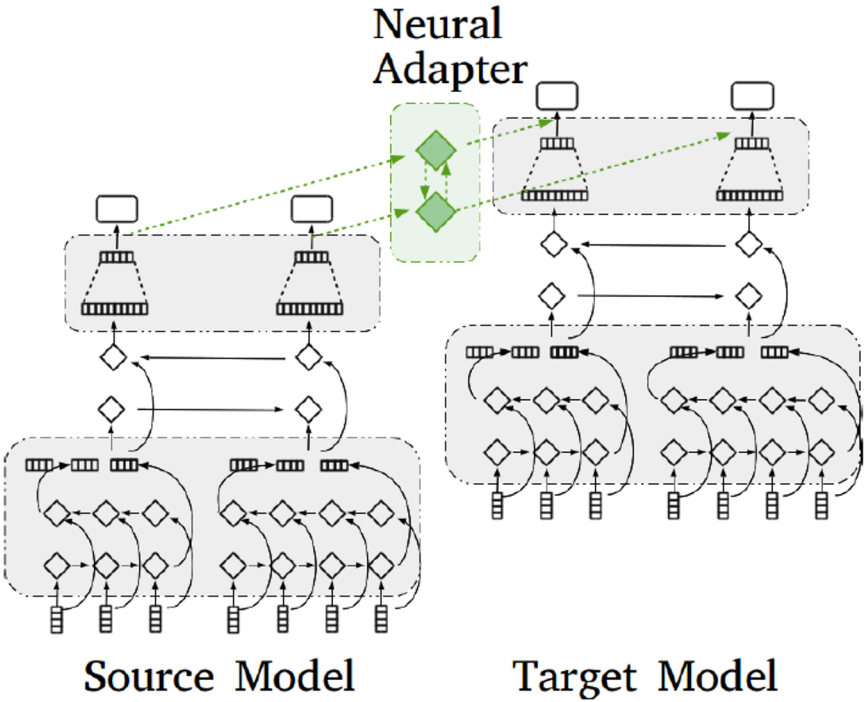
Domain Adaptation in Scrabble

- Two layered Workflow
- Given labeled tagsets from source building tags, predict the labels on unlabeled target building
- Problem Statement- Domain adaptation in the second layer from IR to tagsets



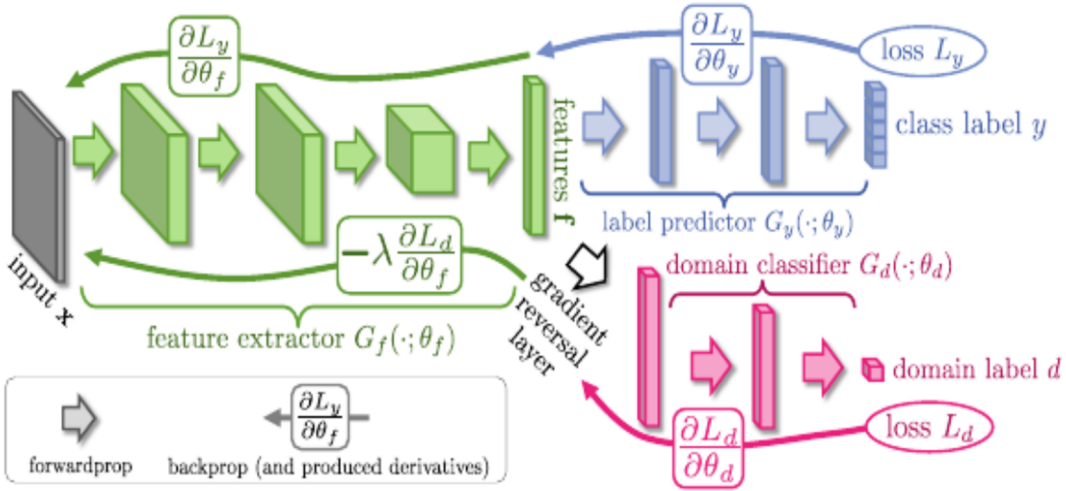
Neural Adapter

- Neural adapter layer(MLP) is used to mitigate the differences between the source and target distributions
- **Pre-training of source model**
- **Parameter transfer**
- **Target model training**



Adversarial Domain Adaptation

- Adversarial training mechanism
- **Feature extractor** - Transformation on both source and target domain
- **Label Classifier** - Predict the final labels
- **Domain Classifier** - Uses gradient reversal layer



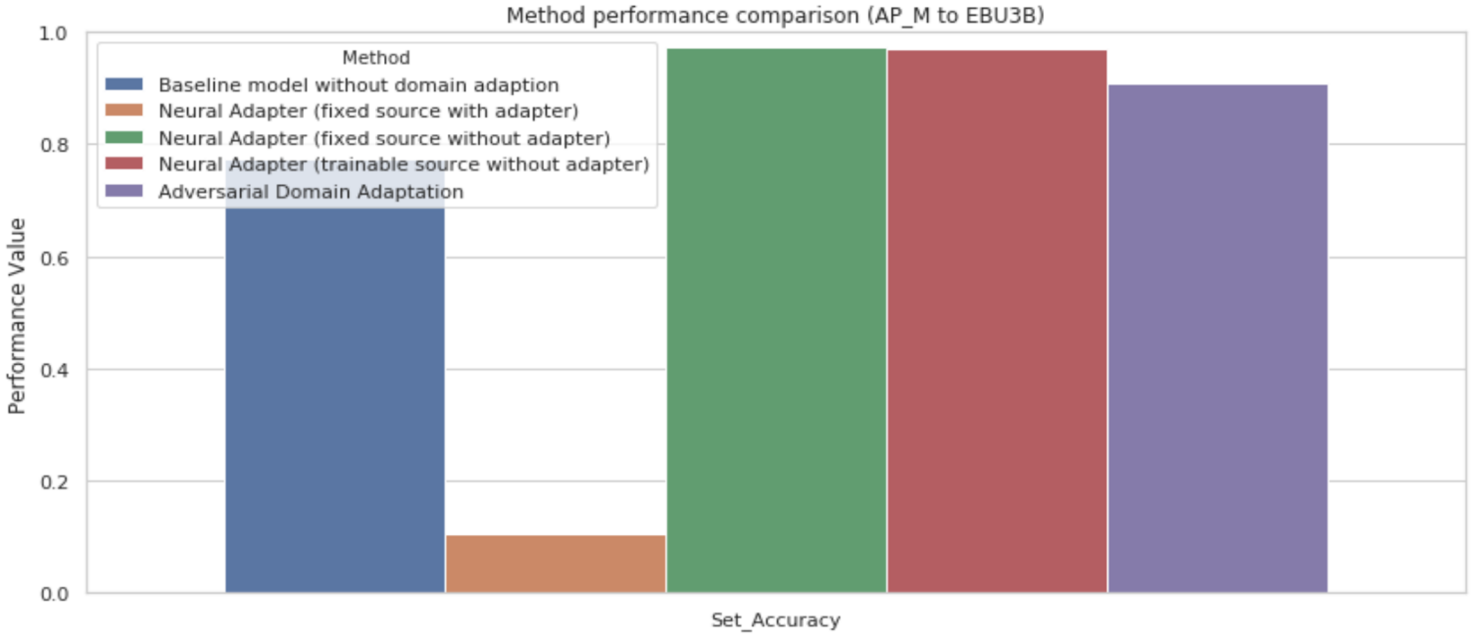
Implementation Techniques

- Preprocessing of the data - obtain samples, domain index and CountVectorizer
- **Neural Adapter with fixed source MLP and adapter layer** -
  Source layers are non trainable, ‘relu’ activation, ‘rmsprop’ optimization, 100 epochs
- **Neural Adapter with fixed source MLP and no adapter layer** -
  Source layers are non trainable, ‘relu’ activation, ‘rmsprop’ optimization, 100 epochs
- **Neural Adapter with variable source MLP and no adapter layer** -
  Source layers are trainable, ‘relu’ activation, ‘rmsprop’ optimization, 100 epochs
- **Adversarial Domain Adaptation** -
  Functional Keras to build and connect three separate models
  ‘relu’ activation, ‘rmsprop’ optimization, 200 epochs

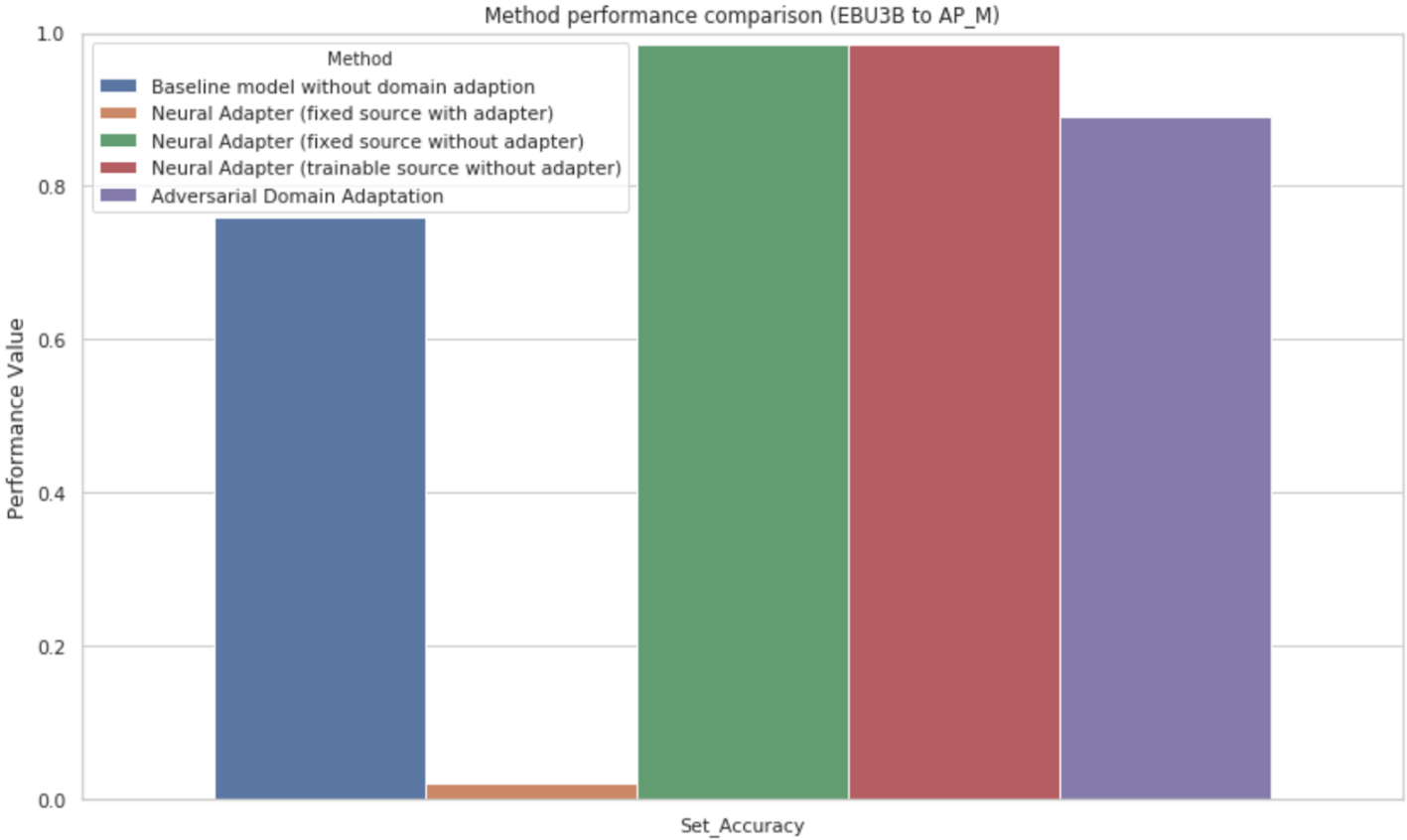
Results

- Evaluation metric - **Set Accuracy** (Sum of all srcid jaccard indices)

```
jaccard = len(pred.intersection(true)) / len(pred.union(true))
```



Results


Method performance comparison (EBU3B to AP_M)
Method
Baseline model without domain adaption
Neural Adapter (fixed source with adapter)
Neural Adapter (fixed source without adapter)
Neural Adapter (trainable source without adapter)
Adversarial Domain Adaptation
Performance Value
1.0
0.8
0.6
0.4
0.2
0.0
Set_Accuracy

Key Inferences

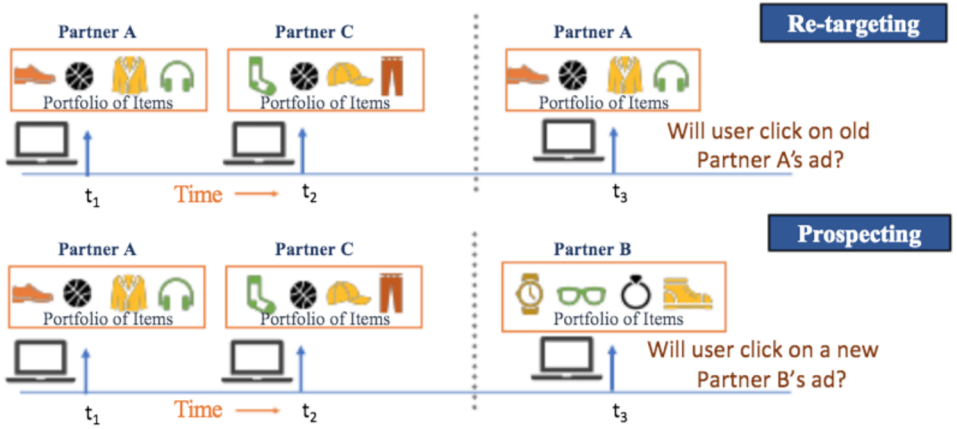
- The domain shifts between the two buildings produced consistent results in both directions.
- For neural adapter, the performance did not vary much if the source layer was made trainable or not. The adapter layer did not perform well due to some bias to zero values while training or possible overfitting.
- The adversarial domain adaptation method performs better than baseline model and results in good predictions due to efficient domain classification
- But this method does not cross 0.9 set accuracy possibly due to the following two reasons -
  Gradient reversal layer leads to vanishing gradient problem
  Same weights are used for source and target transformations

Business Applications of Domain Adaptation

- **Digital Advertisements** -
  Prospecting platform could use adaptation techniques to adapt re-targeting domain to solve cold-start problem

- **Visual Classification tasks** -
  E-commerce websites to classify product images based on image content

- **Deep Fakes** -
  Classify real image from a fake



Thank You

Ashish Gupta

University of California, San Diego

asg007@ucsd.edu