Domain Adaptation with Adversarial Training and Graph Embeddings



Firoj Alam
@firojalam04

Shafiq Joty[†]

Muhammad Imran
@mimran15

Qatar Computing Research Institute (QCRI), HBKU, Qatar
School of Computer Science and Engineering[†]
Nanyang Technological University (NTU), Singapore[†]

@aidr_qcri

Time Critical Events

Disaster events (earthquake, flood)

Urgent needs for affected people

- Food, water
- Shelter
- Medical assistance
- Donations
- Service and utilities

Information gathering

Info.

Info.

Info.

Information gathering in real-time is the most challenging part

AIDR artificial intelligence for digital response

Humanitarian organizations and local administration need information to help and launch response

OCHA

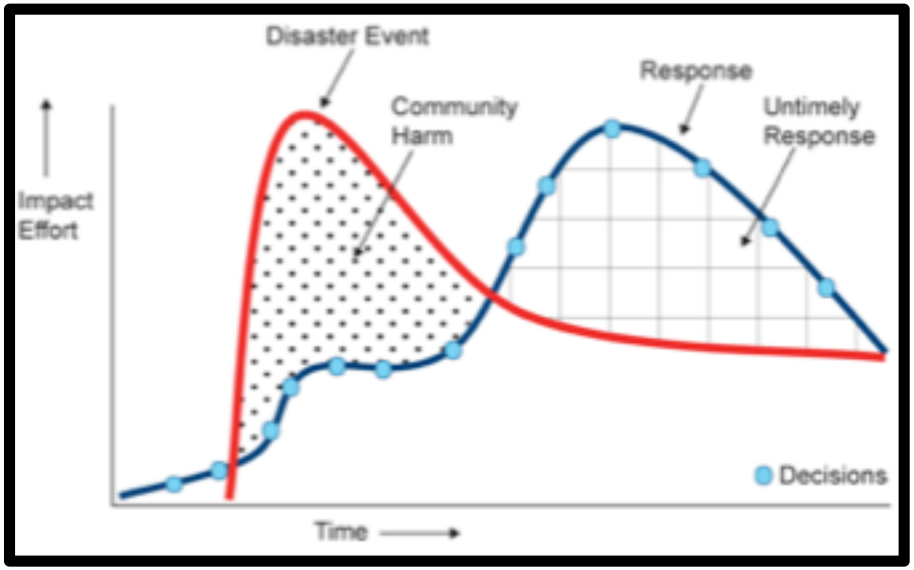
unicef

World Health Organization

Relief operations

Artificial Intelligence for Digital Response (AIDR)

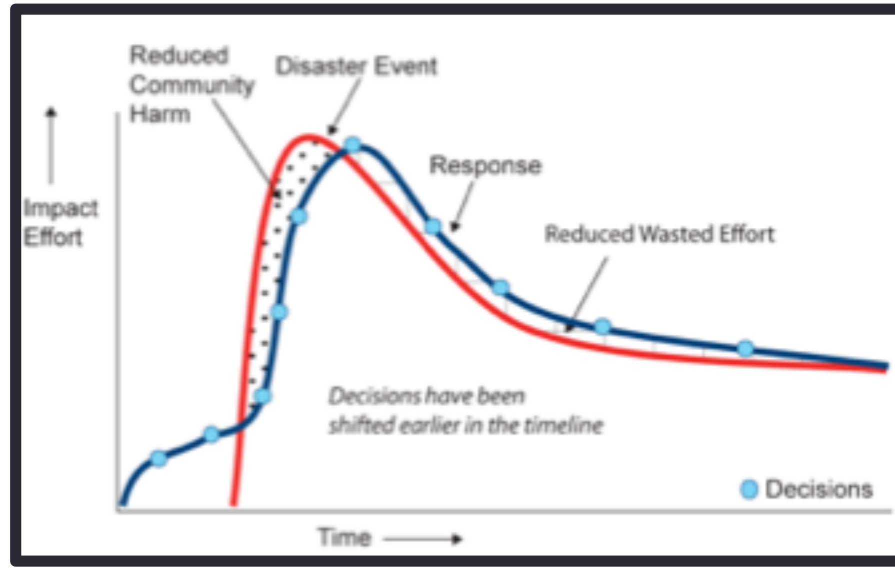
Response time-line today



- Delayed decision-making
- Delayed crisis response

Target

Response time-line our target



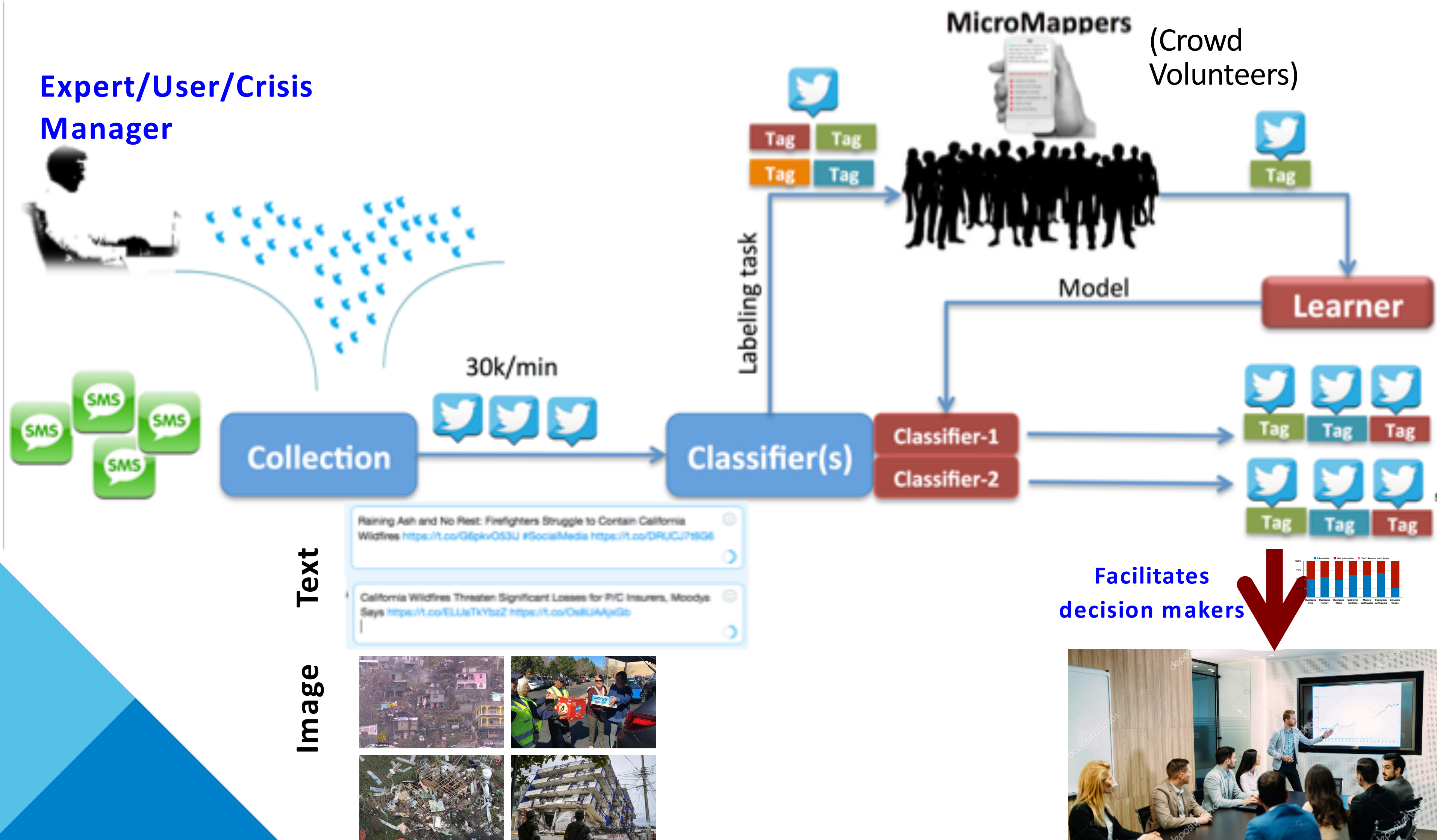
- Early decision-making
- Rapid crisis response

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Artificial Intelligence for Digital Response

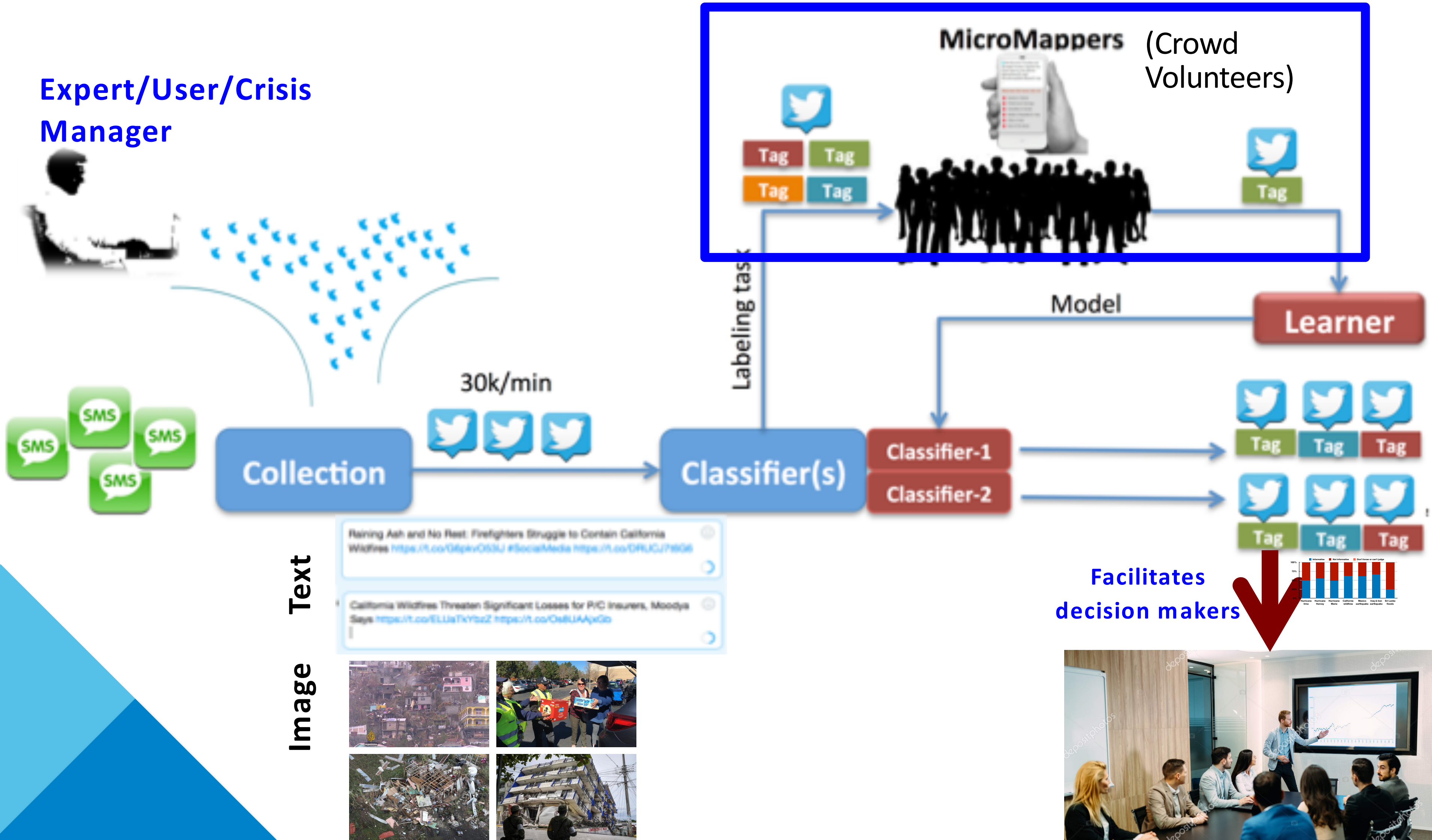
AIDR artificial intelligence for digital response

http://aidr.qcri.org



Artificial Intelligence for Digital Response

http://aidr.qcri.org



Artificial Intelligence for Digital Response

http://aidr.qcri.org

Expert/User/Crisis Manager

- **Small amount of labeled data and large amount of unlabeled data at the beginning of the event**
- **Labeled data from the past event. Can we use them? What about domain shift?**

Facilitates decision makers

Our Solutions/Contributions

- How to use large amount of unlabeled data and small amount of labeled data from the same event?

  $\Rightarrow$ Graph-based semi-supervised

Our Solutions/Contributions

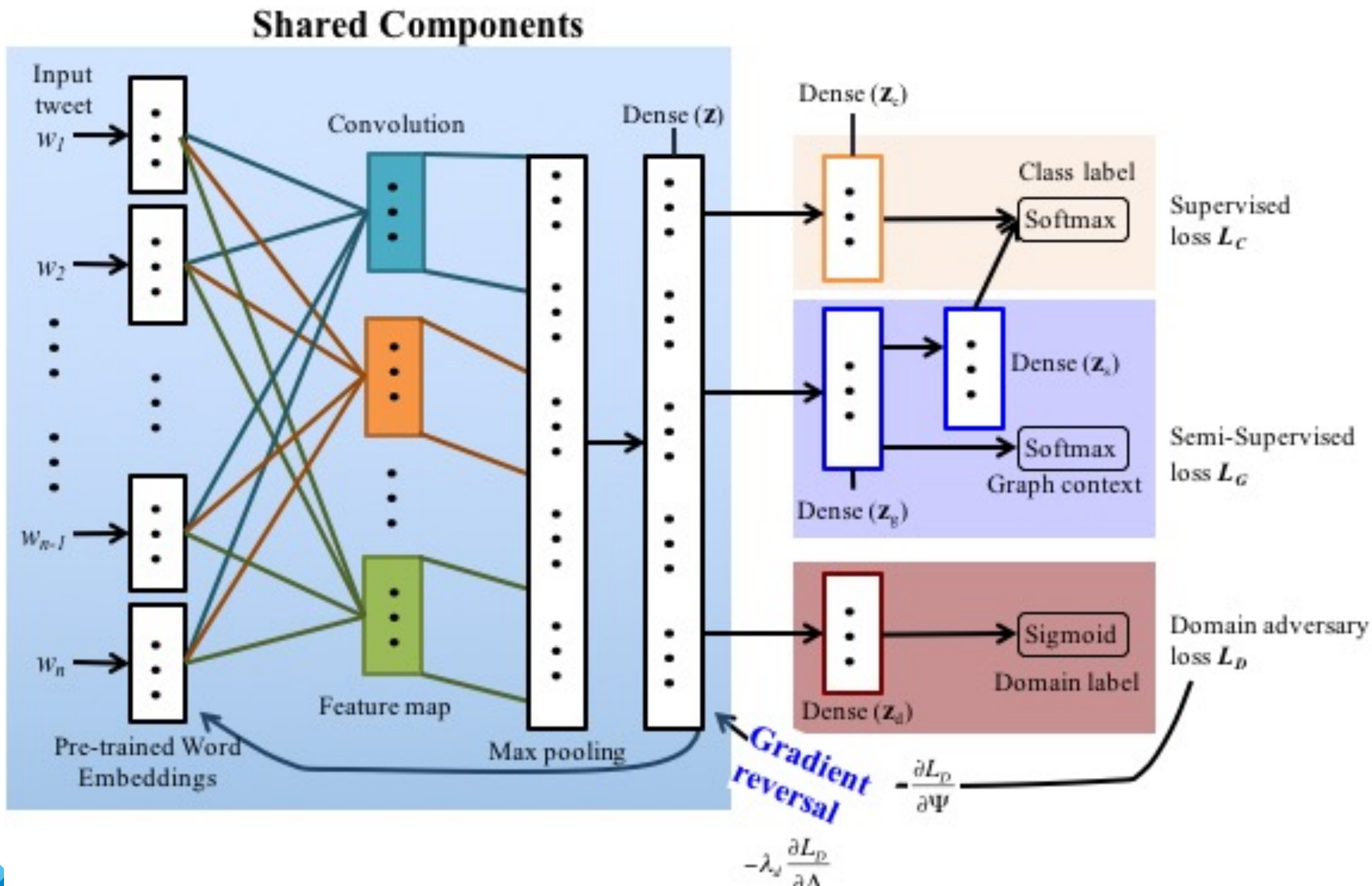
- How to use large amount of unlabeled data and small amount of labeled data from the same event?

  $\Rightarrow$ Graph-based semi-supervised

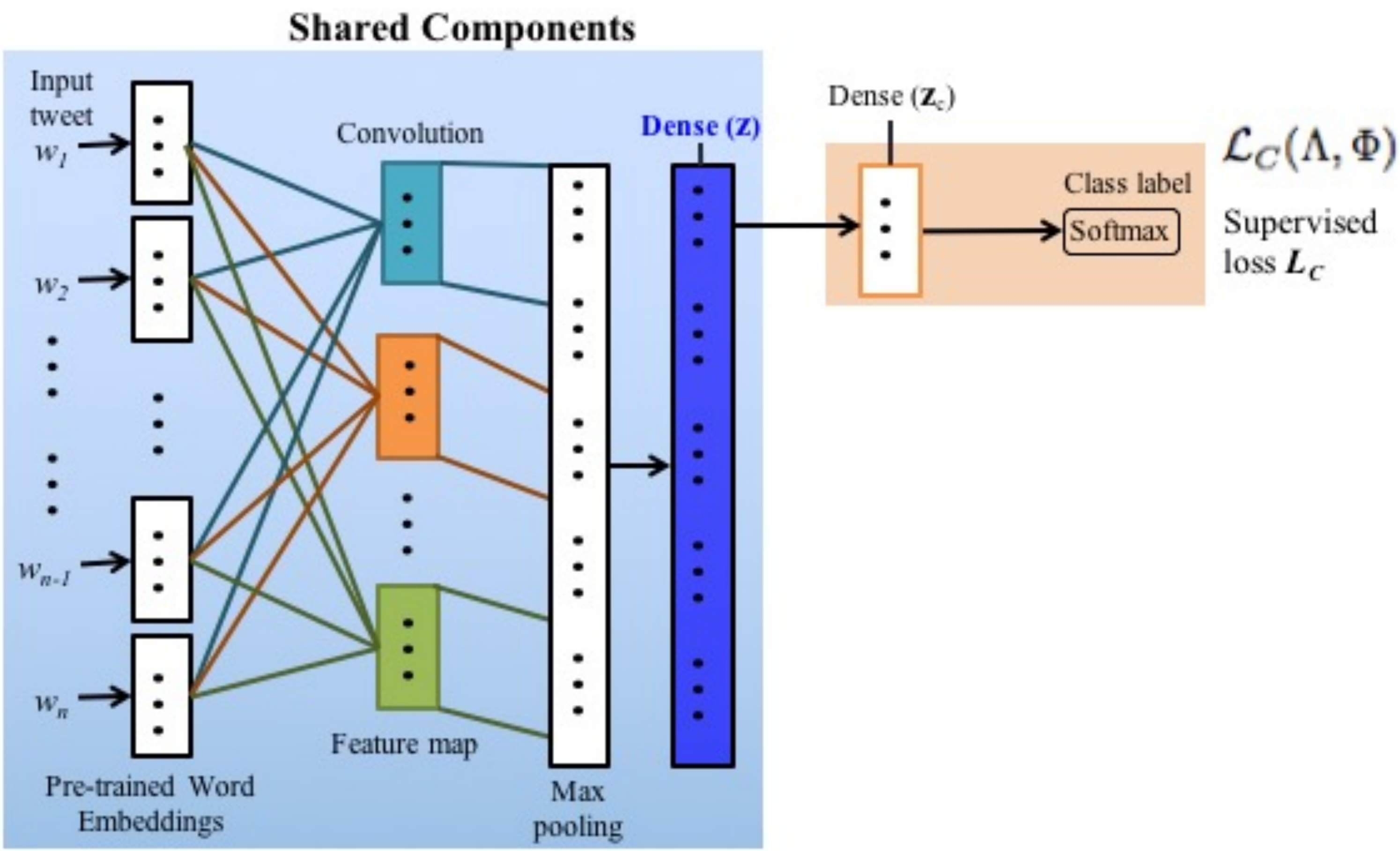
- How to transfer knowledge from the past events

  => Adversarial domain adaptions

Domain Adaptation with Adversarial Training and Graph Embeddings

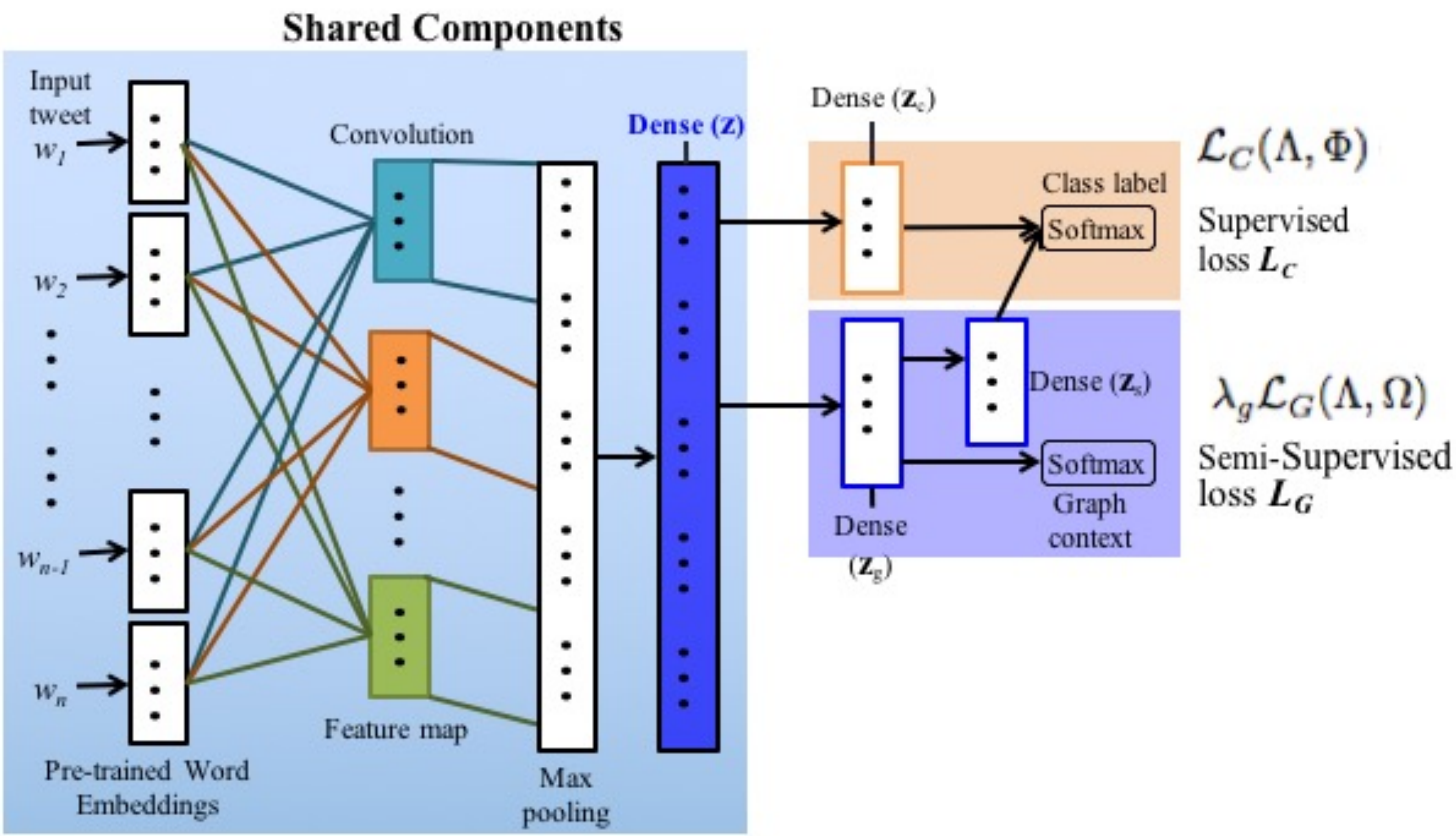


Supervised Learning



Semi-Supervised Learning

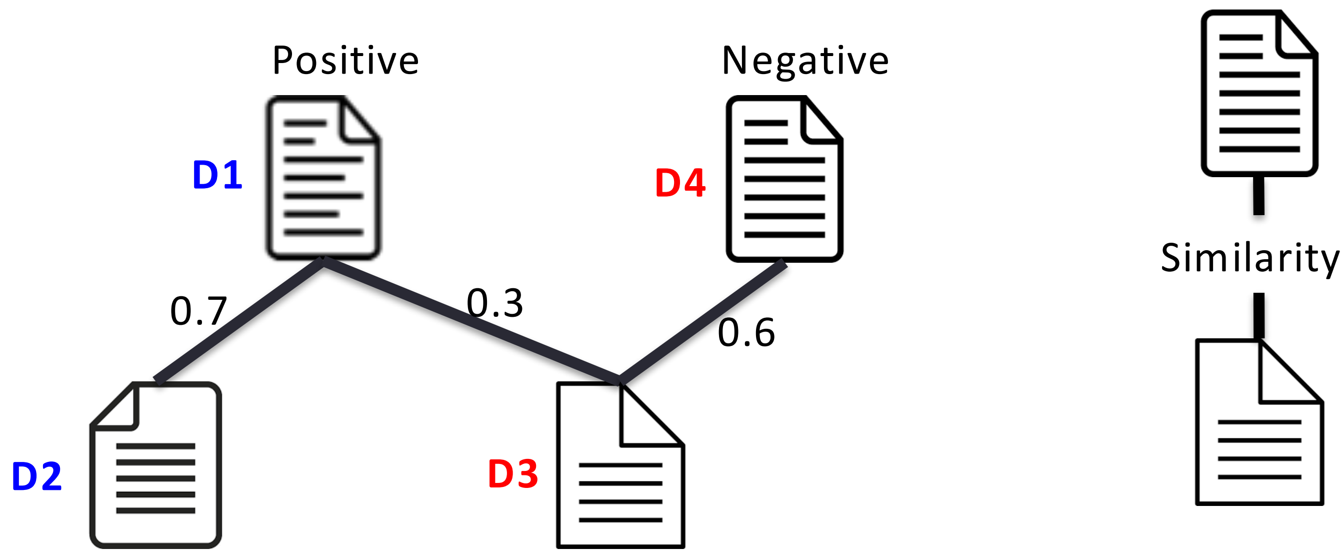
- **Semi-Supervised component**



Semi-Supervised Learning

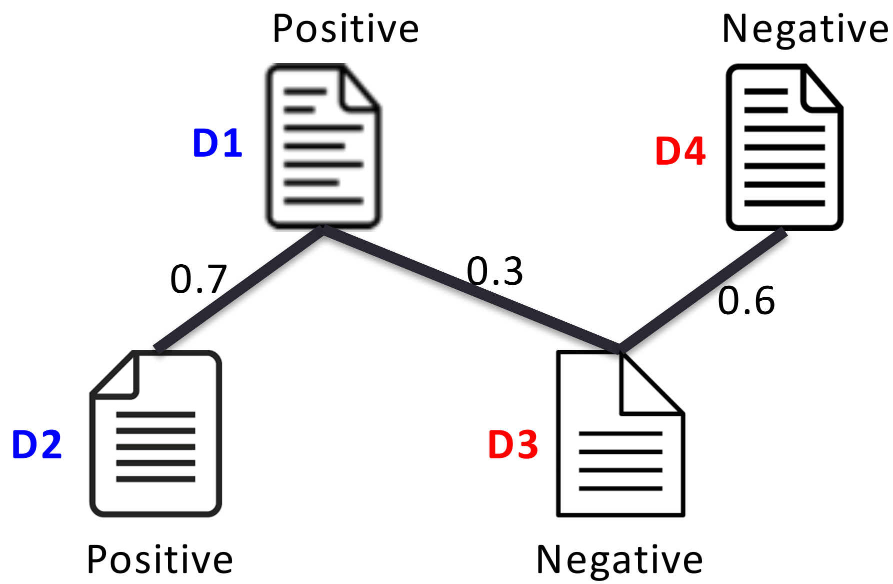
- ***L***: number of labeled instances ($\mathbf{x}_{1:L}$, $\mathbf{y}_{1:L}$)
- ***U***: number of unlabeled instances ($\mathbf{x}_{L+1:L+U}$)
- Design a classifier $f: x \rightarrow y$

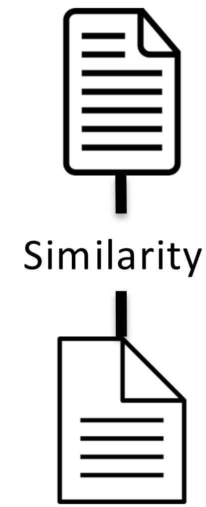
Graph based Semi-Supervised Learning



Assumption: If two instances are similar according to the graph, then class labels should be similar

Graph based Semi-Supervised Learning





Two Steps:

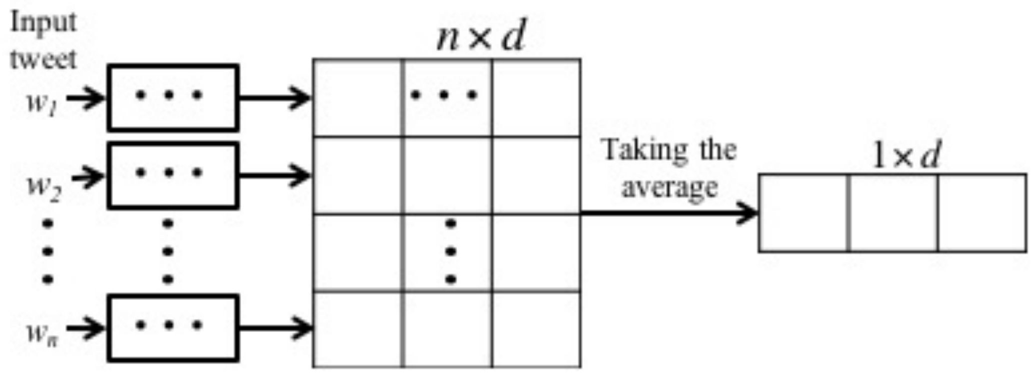
- Graph Construction
- Classification

Graph based Semi-Supervised Learning

- **Graph Representation**
  - Nodes: Instances (labeled and unlabeled)
  - Edges: n x n similarity matrix
  - Each entry $a_{i,j}$ indicates a similarity between instance *i* and *j*

Graph based Semi-Supervised Learning

- **Graph Construction**
  - We construct the graph using k-nearest neighbor (k=10)
    - *Euclidian distance*
    - Requires n(n-1)/2 distance computation
    - *K-d tree data structure to reduce the computational complexity O(logN)*
    - ***Feature Vector:*** *taking the averaging of the word2vec vectors*



Graph based Semi-Supervised Learning

- **Semi-Supervised component: Loss function**

$$\mathcal{L}(\Lambda, \Phi, \Omega) = \mathcal{L}_C(\Lambda, \Phi) + \lambda_g \mathcal{L}_G(\Lambda, \Omega)$$

Graph context loss

$$\mathcal{L}_G(\Lambda, \Omega) = -\frac{1}{L_s + U_s} \sum_{i=1}^{L_s+U_s} \mathbb{E}_{(j,\gamma)} \log \sigma \left(\gamma C_j^T \mathbf{z}_g(i) \right)$$ (Yang et al., 2016)

Learns the internal representations (**embedding**) by predicting a node in the graph context

Graph based Semi-Supervised Learning

- **Semi-Supervised component: Loss function**

$$\mathcal{L}_G(\Lambda, \Omega) = -\frac{1}{L_s + U_s} \sum_{i=1}^{L_s+U_s} \mathbb{E}_{(j,\gamma)} \log \sigma \left(\gamma C_j^T \mathbf{z}_g(i) \right)$$ (Yang et al., 2016)

Two types of context

1. Context is based on the graph to encode structural (distributional) information

Graph based Semi-Supervised Learning

- **Semi-Supervised component: Loss function**

$$\mathcal{L}_G(\Lambda, \Omega) = -\frac{1}{L_s + U_s} \sum_{i=1}^{L_s+U_s} \mathbb{E}_{(j,\gamma)} \log \sigma \left(\gamma C_j^T \mathbf{z}_g(i) \right)$$ (Yang et al., 2016)

Two types of context

1. Context is based on the graph to encode structural (distributional) information
2. Context is based on the labels to inject label information into the embeddings

Graph based Semi-Supervised Learning

- **Semi-Supervised component: Loss function**

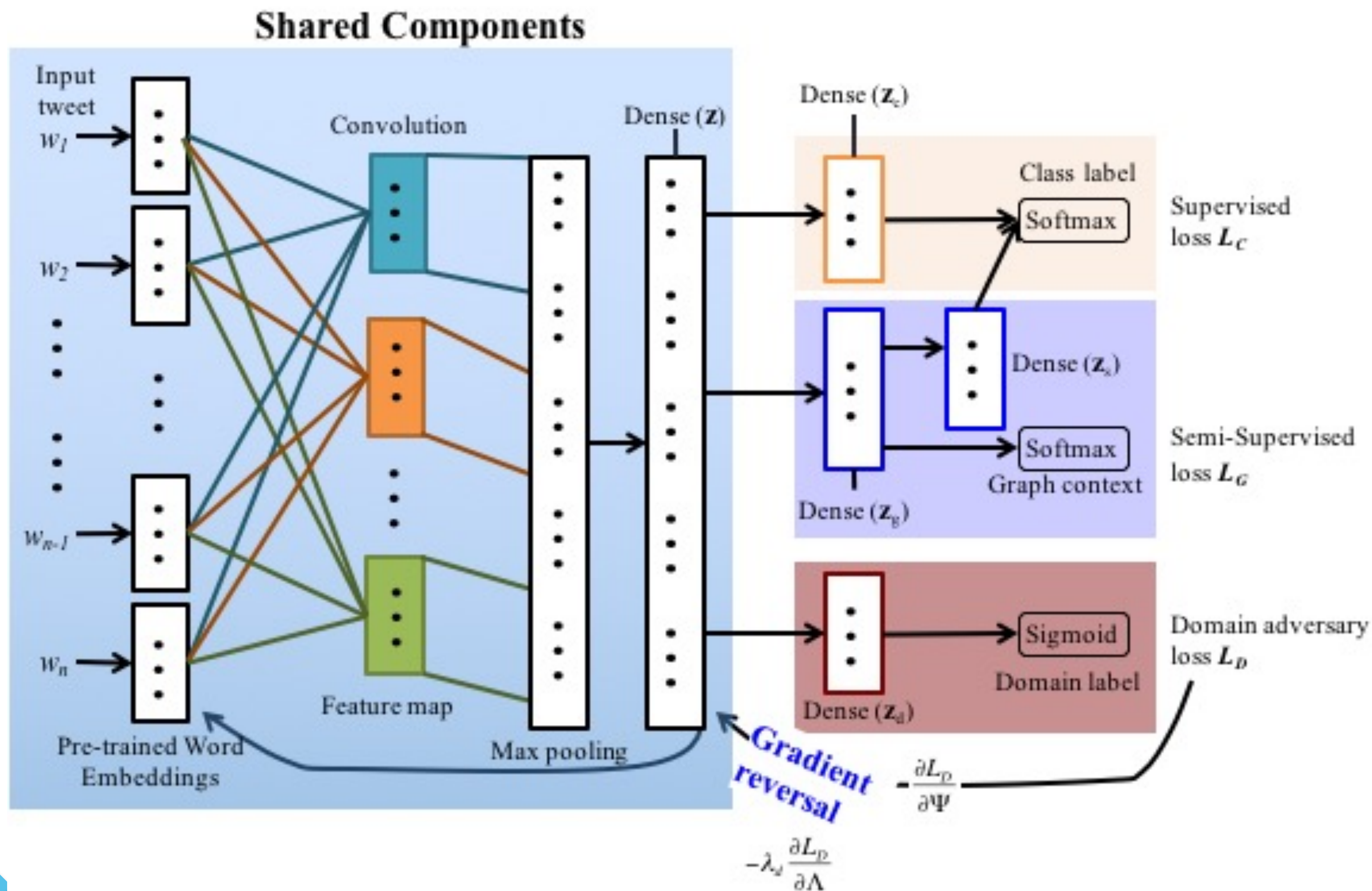
$$\mathcal{L}(\Lambda, \Phi, \Omega) = \mathcal{L}_C(\Lambda, \Phi) + \lambda_g \mathcal{L}_G(\Lambda, \Omega)$$

$\Lambda = \{U, V\}$ Convolution filters and dense layer parameters

$\Phi = \{V_c, W\}$ Parameters specific to the supervised part

$\Omega = \{V_g, C\}$ Parameters specific to the semi-supervised part

Domain Adaptation with Adversarial Training and Graph Embeddings



Domain Adaptation with Adversarial Training

Domain discriminator is defined by:

$$\hat{\delta} = p(d = 1|\mathbf{t}, \Lambda, \Psi) = \mathrm{sigm}(\mathbf{w}_d^T \mathbf{z}_d)$$

Negative log probability of the discriminator loss:

$$\mathcal{J}_i(\Lambda, \Psi) = -d_i \log \hat{\delta} - (1 - d_i) \log \left(1 - \hat{\delta}\right)$$

Domain adversary loss is defined by:

$$\mathcal{L}_D(\Lambda, \Psi) = -\frac{1}{L_s + U_s} \sum_{i=1}^{L_s+U_s} \mathcal{J}_i(\Lambda, \Psi) - \frac{1}{U_t} \sum_{i=1}^{U_t} \mathcal{J}_i(\Lambda, \Psi)$$

$d \in \{0,1\}$ represents the domain of the input tweet **t**

$\Lambda = \{U, V\}$ Convolution filters and dense layer parameters

$\Psi = \{V_d, w_d\}$ Parameters specific to the domain discriminator part

Domain Adaptation with Adversarial Training and Graph Embeddings

- **Combined loss**

$$\mathcal{L}(\Lambda, \Phi, \Omega, \Psi) = \underbrace{\overbrace{\mathcal{L}_C(\Lambda, \Phi)}^{\text{Supervised}} + \lambda_g \mathcal{L}_G(\Lambda, \Omega)}_{\text{Semi-Supervised}} + \overbrace{\lambda_d \mathcal{L}_D(\Lambda, \Psi)}^{\text{Domain adversarial loss}}$$

We seek parameters that minimizes the classification loss of the class labels and maximizes domain discriminator loss

$$\theta^* = \underset{\Lambda,\Phi,\Omega}{\operatorname{argmin}} \max_{\Psi} \mathcal{L}(\Lambda, \Phi, \Omega, \Psi)$$

$\Lambda = \{U, V\}$	Convolution filters and dense layer parameters
$\Phi = \{V_c, W\}$	Parameters specific to the supervised part
$\Omega = \{V_g, C\}$	Parameters specific to the semi-supervised part
$\Psi = \{V_d, w_d\}$	Parameters specific to the domain discriminator part

Model Training

Algorithm 1: Model Training with SGD

Input : data $\mathcal{D}_S^l$, $\mathcal{D}_S^u$, $\mathcal{D}_T^u$; graph G
Output: learned parameters $\theta = \{\Lambda, \Phi\}$
1. Initialize model parameters $\{E, \Lambda, \Phi, \Omega, \Psi\}$;
2. **repeat**

```
// Semi-supervised
```

for *each batch sampled from* $p(j, \gamma | i, \mathcal{D}_S^l, \mathcal{D}_S^u, G)$ **do**
- $a)$ Compute loss $\mathcal{L}_G(\Lambda, \Omega)$
- $b)$ Take a gradient step for $\mathcal{L}_G(\Lambda, \Omega)$;

end

```
// Supervised & domain adversary
```

for *each batch sampled from* $\mathcal{D}_S^l$ **do**
- $a)$ Compute $\mathcal{L}_C(\Lambda, \Phi)$ and $\mathcal{L}_D(\Lambda, \Psi)$
- $b)$ Take gradient steps for $\mathcal{L}_C(\Lambda, \Phi)$ and $\mathcal{L}_D(\Lambda, \Psi)$;

end

```
// Domain adversary
```

for *each batch sampled from* $\mathcal{D}_T^u$ **do**
- $a)$ Compute $\mathcal{L}_D(\Lambda, \Psi)$
- $b)$ Take a gradient step for $\mathcal{L}_D(\Lambda, \Psi)$;

end

until *convergence*;

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Corpus

- **Collected during:**
  - 2015 Nepal earthquake
  - 2013 Queensland flood
- A small part of the tweets has been annotated using crowdflower
  - **Relevant:** injured or dead people, infrastructure damage, urgent needs of affected people, donation requests
  - **Irrelevant:** otherwise

Dataset	Relevant	Irrelevant	Train (60%)	Dev (20%)	Test (20%)
Nepal earthquake	5,527	6,141	7,000	1,167	3,503
Queensland flood	5,414	4,619	6,019	1,003	3,011

Unlabeled Instances

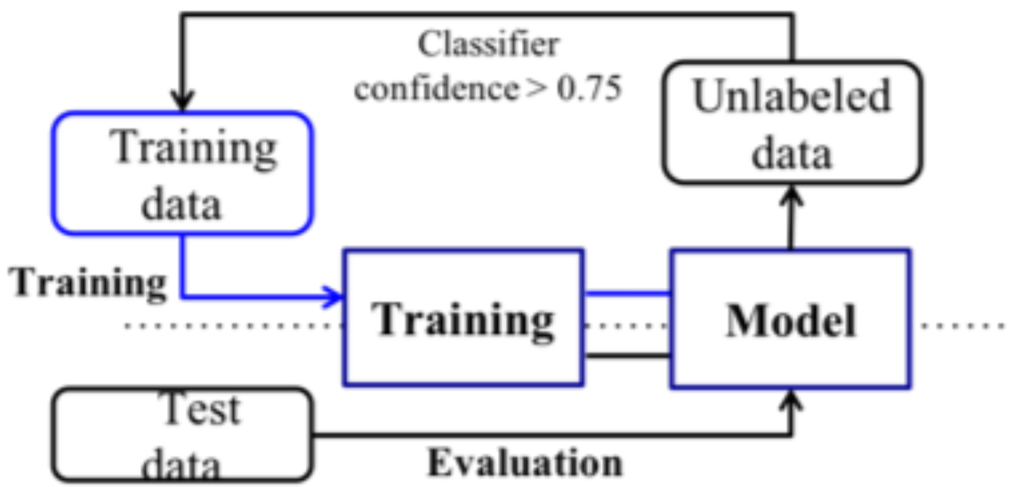
Nepal earthquake: 50K
Queensland flood: 21K

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Experiments and Results

- **Supervised baseline:**
  - Model trained using Convolution Neural Network (CNN)

- **Semi-Supervised baseline (Self-training):**
  - Model trained using CNN were used to automatically label unlabeled data
  - Instances with classifier confidence >=0.75 were used to retrain a new model



QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Experiments and Results

Semi-Supervised baseline (Self-training)

Experiments	AUC	P	R	F1
Nepal Earthquake				
Supervised	61.22	62.42	62.31	**60.89**
Semi-Supervised (Self-training)	61.15	61.53	61.53	**61.26**
Semi-Supervised (Graph-based)	64.81	64.58	64.63	**65.11**
Queensland Flood				
Supervised	80.14	80.08	80.16	**80.16**
Semi-Supervised (Self-training)	81.04	80.78	80.84	**81.08**
Semi-Supervised (Graph-based)	92.20	92.60	94.49	**93.54**

Experiments and Results

- **Domain Adaptation Baseline (Transfer Baseline):** Trained CNN model on source (an event) and tested on target (another event)

Source	Target	AUC	P	R	F1
In-Domain Supervised Model					
Nepal	Nepal	61.22	62.42	62.31	**60.89**
Queensland	Queensland	80.14	80.08	80.16	**80.16**
Transfer Baseline					
Nepal	Queensland	58.99	59.62	60.03	**59.10**
Queensland	Nepal	54.86	56.00	56.21	**53.63**

Experiments and Results

- **Domain Adaptation**

Source	Target	AUC	P	R	F1
In-Domain Supervised Model					
Nepal	Nepal	61.22	62.42	62.31	**60.89**
Queensland	Queensland	80.14	80.08	80.16	**80.16**
Transfer Baseline					
Nepal	Queensland	58.99	59.62	60.03	**59.10**
Queensland	Nepal	54.86	56.00	56.21	**53.63**
Domain Adversarial					
Nepal	Queensland	60.15	60.62	60.71	**60.94**
Queensland	Nepal	57.63	58.05	58.05	**57.79**

Experiments and Results

Combining all the components of the network

Source	Target	AUC	P	R	F1
In-Domain Supervised Model					
Nepal	Nepal	61.22	62.42	62.31	**60.89**
Queensland	Queensland	80.14	80.08	80.16	**80.16**
Transfer Baseline					
Nepal	Queensland	58.99	59.62	60.03	**59.10**
Queensland	Nepal	54.86	56.00	56.21	**53.63**
Domain Adversarial					
Nepal	Queensland	60.15	60.62	60.71	**60.94**
Queensland	Nepal	57.63	58.05	58.05	**57.79**
Domain Adversarial with Graph Embedding					
Nepal	Queensland	66.49	67.48	65.90	**65.92**
Queensland	Nepal	58.81	58.63	59	**59.05**

Summary

- We have seen how graph-embedding based semi-supervised approach can be useful for small labeled data scenario
- How can we use existing data and apply domain adaptation technique
- We propose how both techniques can be combined

Limitation and Future Study

Limitations:

- Graph embedding is computationally expensive
- Graph constructed using **averaged** vector from word2vec
- Explored binary class problem

Future Study

- Convoluted feature for graph construction
- Hyper-parameter tuning
- Domain adaptation: labeled and unlabeled data from target

Thank you!

To get the data: http://crisisnlp.qcri.org/

Please follow us

@aidr_qcri

Firoj Alam, Shafiq Joty, Muhammad Imran. ***Domain Adaptation with Adversarial Training and Graph Embeddings***. ACL, 2018, Melbourne, Australia.

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY