Domain Adaptation with Adversarial Training and Graph Embeddings

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

Firoj Alam
@firojalam04

Shafiq Joty†

Muhammad Imran
@mimran15
Disaster events (earthquake, flood)

Urgent needs for affected people
- Food, water
- Shelter
- Medical assistance
- Donations
- Service and utilities

Information gathering in real-time is the most challenging part

Relief operations

Humanitarian organizations and local administration need information to help and launch response
Artificial Intelligence for Digital Response (AIDR)

Response time-line today

- Delayed decision-making
- Delayed crisis response

Response time-line our target

- Early decision-making
- Rapid crisis response
Artificial Intelligence for Digital Response

http://aidr.qcri.org

Expert/User/Crisis Manager

Facilitates decision makers

MicroMappers (Crowd Volunteers)

30k/min

Collection → Classifier(s) → Classifier-1, Classifier-2 → Model → Learner

Informative
Not informative
Don't know or can't judge

Text

Image

Raining Ash and No Rest: Firefighters Struggle to Contain California Wildfires, https://ti.co/O6FkYV5U #SocialMedia https://ti.co/HjUCJ7r6G6

California Wildfires Threaten Significant Losses for P/C Insurers, Moodya Says https://ti.co/6LiaTbYlZb https://ti.co/3dlUAAjxGb

Facilitates decision makers
Artificial Intelligence for Digital Response

http://aidr.qcri.org

Expert/User/Crisis Manager

Facilitates decision makers

MicroMappers (Crowd Volunteers)

Labeling task

Model

Learner

Collection

Classifier(s)

Classifier-1

Classifier-2

Facilitates decision makers

Text

30k/min

Image
Artificial Intelligence for Digital Response

http://aidr.qcri.org

Expert/User/Crisis Manager

- Small amount of label data and large amount of unlabeled data at the beginning of the event
- Can we use labeled data from the past event? What about domain shift?
Our Solutions/Contributions

• How to use large amount of unlabeled data and small amount of labeled data from the same event?
  ⇒ 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?
  ⇒ Graph-based semi-supervised

• How to transfer knowledge from the past events
  ⇒ Adversarial domain adaptations
Domain Adaptation with Adversarial Training and Graph Embeddings

**Shared Components**

- Input tweet \( w_1 \)
- \( w_2 \)
- \( \ldots \)
- \( w_{n-1} \)
- \( w_n \)
- Pre-trained Word Embeddings
- Convolution
- Feature map
- Max pooling
- Dense (\( z \))

**Supervised loss** \( L_C \)

**Semi-Supervised loss** \( L_G \)

**Domain adversary loss** \( L_D \)

\[ \lambda_d \frac{\partial L_D}{\partial \Psi} \]

\[ \lambda_d \frac{\partial L_D}{\partial A} \]
Supervised Learning

Shared Components

Input tweet $w_t$

$w_2$

$\vdots$

$w_{n-1}$

$w_n$

Pre-trained Word Embeddings

Convolution

Feature map

Max pooling

Dense ($z$)

Dense ($z_c$)

Class label

Supervised loss $L_C$

$\mathcal{L}_C(\Lambda, \Phi)$
Semi-Supervised Learning

• Semi-Supervised component
Semi-Supervised Learning

- $L$: number of labeled instances $(x_{1:L}, y_{1:L})$
- $U$: number of unlabeled instances $(x_{L+1:L+U})$
- Design a classifier $f: x \rightarrow y$
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: $n \times n$ 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 \(O(\log N)\)*
  - **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

\[ \mathcal{L}_G(\Lambda, \Omega) = -\frac{1}{L_s + U_s} \sum_{i=1}^{L_s+U_s} \mathbb{E}_{(j,\gamma)} \log \sigma \left( \gamma C^T_j z_g(i) \right) \] (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

\[
\mathcal{L}_G(\Lambda, \Omega) = - \frac{1}{L_s + U_s} \sum_{i=1}^{L_s + U_s} \mathbb{E}_{(j, \gamma)} \log \sigma \left( \gamma C^T_j z_g(i) \right) 
\] (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

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

**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

Shared Components

- Input tweet \( w_1 \)
- \( w_2 \)
- \( \ldots \)
- \( w_{n-1} \)
- \( w_n \)

Pre-trained Word Embeddings

Convolution

Feature map

Max pooling

Dense (\( z \))

Supervised loss \( L_C \)

Class label

Softmax

Semi-Supervised loss \( L_G \)

Dense (\( z_g \))
Graph context

Sigmoid

Domain adversary loss \( L_D \)

Dense (\( z_d \))
Domain label

\( \lambda_d \left( \frac{\partial L_D}{\partial \Psi} \right) \)

\( \frac{\partial L_D}{\partial \Lambda} \)
Domain Adaptation with Adversarial Training

Domain discriminator is defined by:

$$\hat{\delta} = p(d = 1|t, \Lambda, \Psi) = \text{sigm}(w_d^T z_d)$$

Negative log probability of the discriminator loss:

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

Domain adversary loss is defined by:

$$L_D(\Lambda, \Psi) = -\frac{1}{L_s + U_s} \sum_{i=1}^{L_s+U_s} J_i(\Lambda, \Psi) - \frac{1}{U_t} \sum_{i=1}^{U_t} J_i(\Lambda, \Psi)$$

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

\[ \mathcal{L}(\Lambda, \Phi, \Omega, \Psi) = \mathcal{L}_C(\Lambda, \Phi) + \lambda_g \mathcal{L}_G(\Lambda, \Omega) + \lambda_d \mathcal{L}_D(\Lambda, \Psi) \]

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

\[ \theta^* = \arg\min_{\Lambda, \Phi, \Omega} \max_{\Psi} \mathcal{L}(\Lambda, \Phi, \Omega, \Psi) \]

- \( \Lambda = \{U, V\} \) 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
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
3. until convergence;
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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Relevant</th>
<th>Irrelevant</th>
<th>Train (60%)</th>
<th>Dev (20%)</th>
<th>Test (20%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nepal earthquake</td>
<td>5,527</td>
<td>6,141</td>
<td>7,000</td>
<td>1,167</td>
<td>3,503</td>
</tr>
<tr>
<td>Queensland flood</td>
<td>5,414</td>
<td>4,619</td>
<td>6,019</td>
<td>1,003</td>
<td>3,011</td>
</tr>
</tbody>
</table>

**Unlabeled Instances**
- Nepal earthquake: 50K
- Queensland flood: 21K
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 $\geq 0.75$ were used to retrain a new model
### Experiments and Results

#### Semi-Supervised baseline (Self-training)

<table>
<thead>
<tr>
<th>Experiments</th>
<th>AUC</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nepal Earthquake</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervised</td>
<td>61.22</td>
<td>62.42</td>
<td>62.31</td>
<td>60.89</td>
</tr>
<tr>
<td>Semi-Supervised (Self-training)</td>
<td>61.15</td>
<td>61.53</td>
<td>61.53</td>
<td>61.26</td>
</tr>
<tr>
<td>Semi-Supervised (Graph-based)</td>
<td>64.81</td>
<td>64.58</td>
<td>64.63</td>
<td>65.11</td>
</tr>
<tr>
<td><strong>Queensland Flood</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervised</td>
<td>80.14</td>
<td>80.08</td>
<td>80.16</td>
<td>80.16</td>
</tr>
<tr>
<td>Semi-Supervised (Self-training)</td>
<td>81.04</td>
<td>80.78</td>
<td>80.84</td>
<td>81.08</td>
</tr>
<tr>
<td>Semi-Supervised (Graph-based)</td>
<td>92.20</td>
<td>92.60</td>
<td>94.49</td>
<td>93.54</td>
</tr>
</tbody>
</table>
Experiments and Results

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

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>AUC</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In-Domain Supervised Model</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nepal</td>
<td>Nepal</td>
<td>61.22</td>
<td>62.42</td>
<td>62.31</td>
<td>60.89</td>
</tr>
<tr>
<td>Queensland</td>
<td>Queensland</td>
<td>80.14</td>
<td>80.08</td>
<td>80.16</td>
<td>80.16</td>
</tr>
<tr>
<td><strong>Transfer Baseline</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nepal</td>
<td>Queensland</td>
<td>58.99</td>
<td>59.62</td>
<td>60.03</td>
<td>59.10</td>
</tr>
<tr>
<td>Queensland</td>
<td>Nepal</td>
<td>54.86</td>
<td>56.00</td>
<td>56.21</td>
<td>53.63</td>
</tr>
</tbody>
</table>
# Experiments and Results

- **Domain Adaptation**

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>AUC</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In-Domain Supervised Model</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nepal</td>
<td>Nepal</td>
<td>61.22</td>
<td>62.42</td>
<td>62.31</td>
<td>60.89</td>
</tr>
<tr>
<td>Queensland</td>
<td>Queensland</td>
<td>80.14</td>
<td>80.08</td>
<td>80.16</td>
<td>80.16</td>
</tr>
<tr>
<td><strong>Transfer Baseline</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nepal</td>
<td>Queensland</td>
<td>58.99</td>
<td>59.62</td>
<td>60.03</td>
<td>59.10</td>
</tr>
<tr>
<td>Queensland</td>
<td>Nepal</td>
<td>54.86</td>
<td>56.00</td>
<td>56.21</td>
<td>53.63</td>
</tr>
<tr>
<td><strong>Domain Adversarial</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nepal</td>
<td>Queensland</td>
<td>60.15</td>
<td>60.62</td>
<td>60.71</td>
<td>60.94</td>
</tr>
<tr>
<td>Queensland</td>
<td>Nepal</td>
<td>57.63</td>
<td>58.05</td>
<td>58.05</td>
<td>57.79</td>
</tr>
</tbody>
</table>
## Experiments and Results

Combining all the components of the network

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