Unsupervised Domain Adaptation with Global and Local Graph Neural Networks in Limited Labeled Data Scenario

Application to Disaster Management

Samujjwal Ghosh¹, Subhadeep Maji², and Maunendra Sankar Desarkar¹,
¹Indian Institute of Technology Hyderabad, India
²Amazon, India

Identification and categorization of social media posts generated during disasters are crucial to reduce the sufferings of the affected people. However, lack of labeled data is a significant bottleneck in learning an effective categorization system for a disaster. This motivates us to study the problem as unsupervised domain adaptation (UDA) between a previous disaster with labeled data (source) and a current disaster (target). However, if the amount of labeled data available is limited, it restricts the learning capabilities of the model. To handle this challenge, we utilize limited labeled data along with abundantly available unlabeled data, generated during a source disaster to propose a novel two-part graph neural network. The first-part extracts domain-agnostic global information by constructing a token level graph across domains and the second-part preserves local instance-level semantics. In our experiments, we show that the proposed method outperforms state-of-the-art techniques by 2.74% weighted F₁ score on average on two standard public dataset in the area of disaster management. We also report experimental results for granular actionable multi-label classification datasets in disaster domain for the first time, on which we outperform BERT by 3.00% on average w.r.t weighted F₁. Additionally, we show that our approach can retain performance when very limited labeled data is available. We plan to release our code publicly.

Index Terms—Domain Adaptation, Graph Neural Networks, Text Classification, Disaster Management

I. INTRODUCTION

Identifying relevant and actionable information from the data generated during a disaster can help mitigate the sufferings of the affected people. However, various individuals, government and non-government agencies post about their arrangements, activities and intents towards extending help in the rescue and relief operations. Timely extraction and identification of such actionable information from social media can be beneficial for disaster relief [1]–[3]. Table I presents an example tweet posted during the 2015 Nepal earthquake and the different actionable classes that it corresponds to. The example indicates that the generated posts can contain useful, actionable information, and can be of multi-label in nature. However, a large portion of generated data are sentiments of people towards the disaster. Filtering and categorizing the small percentage of actionable posts from this large pool is challenging. One primary reason for this is the difficulty in obtaining a sufficiently large number of annotations on time to assist development of a classifier.

In this work, we focus on the problem of identification and categorization of actionable social media posts when no labeled data is available from the current disaster. Unsupervised Domain Adaptation (UDA) may be utilized in such a scenario if labeled data is available from a prior disaster is available [5]–[7]. However, even when labeled data from a past disaster is available, the amount of labeled data is generally minimal as annotating posts based on the actionable information is expensive. Lack of labeled data might restrict learning a classifier for any effective representation of domain shift.

To overcome these challenges, we propose an UDA approach based on a cross-domain graph construction which utilizes abundantly available unlabeled data from both past (source) and current (target) disaster. To this effect, we propose a novel two-part (Global and Local) graph-based formalism. The Global-part learns a domain-agnostic feature representation for tokens using a large token graph constructed from labeled data from source domain along with unlabeled data from both the domains to identify analogous terms between the domains and brings them closer. The Local-part operates per instance, preserves the local instance-level semantics by learning contextual representations of the posts. We combine these representations and learn a classifier that generalizes across domains. We refer to our approach as Global and Local Graph Neural Network (GLEN).

In our experiments, we compare GLEN against SOTA methods for UDA in the area of disaster management. We
We effectively utilize the unlabeled data from both source and target domains for UDA in the area of disaster management with a novel two-part graph neural network based framework. We investigate the multi-label setting in UDA for the first time in disaster management and show that our model performs competitively in the multi-label setting as well. We show significant improvement for UDA over SOTA methods and BERT on a number of datasets. We achieve on average 3.74% improvement over SOTA methods and 1.21% improvement over BERT.

II. RELATED WORK

A. Unsupervised Domain Adaptation (UDA)

UDA is an ideal task for disaster management due to the inherent lack of labeled data in target domain. Existing UDA approaches can be categorized into four major types:

- Feature projection [5], [8], [9]
- Instance re-weighting [10]–[12]
- Pivot feature centric [13], [14]
- Domain Adversarial / Gradient Reversal based [6], [15]–[19]

Feature projection signifies bringing the features of source and target domain to a joint latent space. [8] used stacked autoencoders to learn domain adaptive feature representations for sentiment analysis. In instance re-weighting, source domain samples are identified which are similar to target domain, and the model is re-trained with weighted instances from source domain. However, [20] showed that instance re-weighting might suffer from poor performance if there is limited similarity between the domains. In such a scenario, the model assigns large weights to a few samples resulting in a bias towards those samples. This limitation is addressed in [10] by introducing sample selection variance in instance re-weighting. Sample selection variance ensures that weights are distributed over a larger population of source data and do not rely on a few highly relevant instances. Instance re-weighting and feature projection is combined in [10]. Features are projected such that class differences are maximized, which reduces misclassification. Pseudo labels are then learned for unlabeled target samples. Finally, the classifier is updated using the target domain data along with their pseudo labels as true classes. Instance re-weighting approaches are computationally expensive as the model needs to be re-trained. Pivot Based Language Modelling (PBLM) approaches work by predicting if a word is pivot between source and target domains. [13] applied PBLM on sentiment classification task. However, finding a large number of pivots is challenging. To mitigate the challenge, the authors proposed Task Refinement Learning (TRL-PBLM) in their followup work [14] by dividing the pivots into subsets and predicting the subset instead of the pivot directly. Domain Adversarial approaches are frequent for (un)supervised domain adaptation. [15]–[18] use gradient reversal for news classification, stance detection, duplicate question detection and machine reading comprehension respectively. Main objective of these approaches is to maximize domain difference using a gradient reversal layer over domain classification loss. [15] uses Domain-Adversarial Graph Neural Network (DAGNN) to represent both source and target domain documents as instance graphs to capture local context only in both domains.

B. UDA for Disaster Response

An additional challenge specific to disasters is the difficulty of obtaining labeled data in source domain. Annotating large amount of examples based on the type of actionable information is expensive. Domain adaptive approaches in disaster domain should consider this constraint along with non-availability of labeled target domain data. As the amount of labeled data from source domain is limited, UDA approaches in disaster utilizes unlabeled source data. Whereas, majority of UDA approaches discussed in previous Subsection II-A do not utilize source domain unlabeled data.

To classify tweets relevant to a disaster, [6] utilized labeled and unlabeled data from source domain along with unlabeled data from target domain. In their proposed end-to-end approach, adversarial learning with gradient reversal and graph convolutional networks is utilized to achieve domain adaptation between a pair of source and target disaster. A document-level graph was constructed using k-nearest neighbours of Word2Vec [21] vectors. However, document level embedding might fail to capture the token-level distribution shift between source and target domain. SOTA performance in disaster scenario was achieved recently in [5] where the authors use a domain reconstruction classification network (DRCN). DRCN reduces the covariate shift by reconstructing the target domain data with an LSTM based autoencoder. Reconstruction based approaches do not utilize token-level information. Although, several approaches in the literature explore domain adaptation setting for disaster relief, majority of disaster relief approaches were limited to predicting binary relevance of examples only. To overcome these challenges in disaster domain, we propose a Graph Neural Network based approach which explicitly models both document-level local contexts to maximize class separation along with token-level global context to minimize the domain difference. We achieve this by using labeled source data for local context and unlabeled data from source and target domains for token-level global context. We apply our model on multiple datasets with varying amount of labeled data in source domain.
Although availability of source domain labeled data is limited in disaster domain, this challenge has not been investigated before in disaster domain. We experiment on real world multi-label disaster tweet datasets where the amount of available labeled data in source domain is very limited (approx 1000 training examples or lower).

III. PROPOSED METHOD

We refer to the past disaster domain as source (S) and current one as target (T). We assume access to labeled data only in S, in the form of \( \{ x, y \} \), where \( y \) is a label corresponding to the instance (e.g. tweet) \( x \). In this setting we would like to learn a classifier in domain T, using labeled (and unlabeled) data from S and using only unlabeled data from T. We refer to the labeled and unlabeled subsets of instances from S as \( S_L \) and \( S_U \) respectively and refer to unlabeled data from T as \( T_U \).

Our model is composed of a two-part (Global and Local) graph based framework, trained in an end-to-end manner using the classification loss on labeled source data. Both these graphs are defined on the tokens available in the data. The Global-part helps in learning domain agnostic feature representations of tokens in instances \( x \) using a large weighted graph constructed from tokens in \( S_L \cup S_U \cup T_U \). The edge weights in this graph are constructed based on token co-occurrence information. The Local-part helps in preserving the local per-instance context in token representations using an unweighted graph constructed from the tokens of instance \( x \).

One important point to consider here is that the global graph is weighted and the local graph is unweighted. This is because, the global graph is constructed from all the available posts from source and target domain. Since the number of such posts is large, the token co-occurrence information can be computed from a large sample, and can be indicative of the contextual similarity between the tokens. The same information can be further leveraged in the subsequent steps. On the other hand, the local graph is instance-specific and is constructed from individual posts. Co-occurrence of tokens in a single post (which is a short text) can be spurious, or also can happen by accident. Hence, although we retain the co-occurrence information in the local graph, we do not set any weight for the edges. However, we learn these weights using an attention framework whose parameters are learned during the training process.

We then combine representations of tokens obtained from these two graphs to form unified token representation that are both (a) domain agnostic and (b) retain per-instance level context. Figure 1 illustrates an overview of our approach. Next, we describe the individual components of our model in detail.

A. Global-Token Graph

One of the key challenges in domain adaptation is to mitigate the covariate shift between domains. Formally, covariate shift is the difference between the feature conditional distributions in the two domains, i.e. \( p_S (\phi (x) \mid y) \neq p_T (\phi (x) \mid y) \), where
\( \phi \) is an appropriate featurization of \( x \). As an example, consider the two tweets \( x_1 = \text{“I need water”} \) and \( x_2 = \text{“I need wifi”} \) from two different domains both associated with the label ‘Resources Required’. The example illustrates the difference in kind of resources needed when a disaster (e.g. earthquake) hits two different locations with vastly different demographics and socio-economic status such as developing nation ‘Nepal’ and developed world nation like ‘Italy’.

However, in unsupervised setting, we do not have access to labels in the target domain, and we address the covariate shift approximately by learning a domain agnostic feature representation \( \phi \) of tokens \( t_i \) across domains \( S \) and \( T \). As a concrete example, we would want \( \phi(\text{“I need water”}) \approx \phi(\text{“I need wifi”}) \).

We form a weighted graph \( G = (V,E,P) \). The vertex set \( V \) consists of all the tokens from combined corpus in both domains i.e. \( S_L \cup S_U \cup T_U \). Two tokens \( t \) and \( t' \) are connected by an edge in \( E \) if they co-occur within a context window of an instance in either of \( S_L, S_U \) or \( T_U \). The weight of an edge \( \rho(\hat{t}, \hat{t}') \) is defined as,

\[
\rho(\hat{t}, \hat{t}') = \frac{1}{2} \left( \frac{n_{\hat{t}, \hat{t}'}^S + n_{\hat{t}', \hat{t}'}^S}{n_{\hat{t}}^S + n_{\hat{t}'}^S} + \frac{n_{\hat{t}, \hat{t}'}^T + n_{\hat{t}', \hat{t}'}^T}{n_{\hat{t}}^T + n_{\hat{t}'}^T} \right) \tag{1}
\]

where, \( n_{\hat{t}, \hat{t}'}^{S,T} \) are the co-occurrence counts of \( \hat{t}, \hat{t}' \) within a context window in \( S \) and \( T \) respectively and \( n_{\hat{t}} \) (and \( n_{\hat{t}'} \)) are the number of occurrences of the token \( t \) (and \( t' \)) in the two domains. As these edge weights are estimated from a large corpus, we consider these weights to be reflective of underlying token co-occurrence frequency. The graph \( G \) captures the notion of co-occurrence of tokens across two domains in its neighborhood. As a concrete example, from tweets \( x_1 \) and \( x_2 \), the token nodes \( \text{water} \) and \( \text{wifi} \) would be 2-hop neighbors of each other connected by the node \( \text{need} \) in \( G \). Also, the nodes \( \text{nepal} \) and \( \text{italy} \) are connected via earthquake in Figure 1.

Taking advantage of this neighborhood in \( G \), we define a Graph Convolutional Network (GCN) \([22]\) on \( G \) to learn representations for the token nodes. In GCN, the node representations are learnt using the multi-hop neighborhood of a node. Therefore, GCN learns similar node representations in a neighborhood. Since the neighborhoods in \( G \) are composed of tokens from \( S \) and \( T \), this results in domain agnostic feature representations for tokens in \( x \). Formally, we define a GCN \([22]\) as follows,

\[
H^{(l+1)} = \sigma \left( D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \tag{2}
\]

with \( H^{(0)} = \text{GloVe}(x) \). Here, \( D \) is the diagonal degree matrix of \( G \) and \( A = A + I \), where \( A \) is the adjacency matrix of \( G \). We restrict to 2-hop neighborhood, thus \( l = 0, 1 \).

In our experiments, we will show qualitatively how the GCN based representations improve the GloVe \([23]\) based representations by bringing the tokens, that appear in similar contexts in the two domains, closer to one another.

### B. Local-Example Graph

The global token graph layer discussed in Section III-A builds domain agnostic feature representation of tokens in instances \( x \). However, it does not take into account the local context defined by tokens of \( x \) alone. The recent success of context-dependent word embedding such as BERT \([24]\) in a wide range of NLP tasks indicate the importance of modelling the local context.

To model the local context, we define an instance-level graph \( G_x = (V_x, E_x) \), where \( V_x \) is the set of tokens in an instance \( x \) and are connected by an edge in the edge set \( E_x \) if they appear within a context window of each other. It is noteworthy that \( G_x \) is an unweighted graph because it would be challenging to apply a weighting scheme owing to the local context. Instead, we propose to learn the weights in the form of attention scores on the edges in a Graph Attention Network (GAT) \([25]\) framework. The GAT on the local instance graphs \( G_x \) learns node representations for tokens in instance \( x \) along with the edge attention scores in an end-to-end manner. Formally, we define a \( k \)-layer GAT on \( G_x \) parameterized by \( W^k \) as follows,

\[
h^{(l+1)}_i = \bigoplus_k \sum_{j \in N(i)} \alpha^{k}_{ij} W^k h^{(l)}_j \tag{3}
\]

with \( h^{(0)}_i = \text{GloVe}(x_i) \). Here, \( h^{(l)}_i \) is the \( l \)-th layer representation of token node \( x_i \), \( N(i) \) are tokens belonging to context window around the token \( x_i \) and \( \oplus \) indicates concatenation. We restrict to a 2-layer GAT i.e. \( l = 0, 1 \) and use two attention heads in first layer and a single head in the second layer. We consider the context window around token \( x_i \) as \( [x_i-1 ... x_i+1] \). Figure 1 shows such an instance graph with window size 2 for a tweet posted during 2015 Nepal earthquake.

It is worthwhile to note that GAT is closely related to the self-attention mechanism in Transformer based architectures \([26]\). The key advantage of GAT over transformers in our setting is the fact that GAT’s attention mechanism is restricted to the neighborhood of a token, which in our case is the context window \( [x_i-1 ... x_i+1] \). In comparison, transformer based models learn attention scores over the entire sequence at every position in the sequence and require much larger number of parameters.

### C. Classification Layer

In Section III-A and Section III-B, we focused on defining domain agnostic representations of tokens \( x_i \) and on preserving the local instance context respectively. We finally combine these two representations by concatenating them as follows:

\[
h(x_i) = h^{(2)}_{G_x}(i) \oplus h^{(2)}_{G}(j) \tag{4}
\]

where, \( j \) is the index of the token \( x_j \) in the token graph \( G \). Also, \( h^{(2)}_{G_x} \) and \( h^{(2)}_{G} \) are final layer representations from local instance graph GAT (Eq. (3)) and global token graph GCN (Eq. (2)) respectively. The concatenated representation combines the global domain agnostic and the local context dependent nature of the tokens in instances \( x \).

Using these representations \( h(x_i) \) as input features in a single layer bi-directional LSTM (BiLSTM) network, we learn an end-to-end model. The parameters of the global token graph (the underlying GCN) and local instance graph (the underlying
TABLE II: Stats of NEQ and QFL datasets. The class 1 indicates tweets relevant to disaster and 0 indicates otherwise.

| Dataset | 1  | 0  | Train | Val | Test |
|---------|----|----|-------|-----|------|
| NEQ     | 5527 | 6141 | 7000  | 1166 | 3502 |
| QFL     | 5414 | 4619 | 6019  | 1003 | 3011 |

TABLE III: Class mapping from FIRE16 to SMERP17. Class 5 of FIRE16 was ignored.

| FIRE16 Class name | ID | ID |
|-------------------|----|----|
| Resources Available | 1  | 1  |
| Medical Resources Available | 3  | 1  |
| Resources Required | 2  | 2  |
| Medical Resources Required | 4  | 2  |
| Resources Specific Locations | 5  | 5  |
| Infrastructure Damage & Restoration | 6  | 3  |
| Activities NGOs / Government | 7  | 4  |

TABLE IV: Details of FIRE16 dataset.

| FIRE16 | Train | Val | Test | Class Dist(%) |
|--------|-------|-----|------|---------------|
| 1      | 498   | 55  | 237  | 35.94         |
| 2      | 217   | 24  | 104  | 15.69         |
| 3      | 367   | 41  | 175  | 26.52         |
| 4      | 302   | 34  | 144  | 21.83         |
| # Examples | 957 | 106 | 459 |

TABLE V: Details of SMERP17 dataset.

| SMERP17 | Train | Val | Test | Class Dist(%) |
|---------|-------|-----|------|---------------|
| 1      | 184   | 22  | 76   | 13.60         |
| 2      | 105   | 15  | 46   | 8.00          |
| 3      | 774   | 141 | 393  | 63.09         |
| 4      | 212   | 25  | 80   | 15.29         |
| # Examples | 1159 | 189 | 548 |

IV. EXPERIMENTS

The main empirical focus of our work is to measure the effectiveness of our approach when no labeled data is available from target domain. To this effect, considering an UDA setting, we pair up public datasets from disaster management as source (S) and target (T) for our experiments and supervise our model (refer to Section III-C) on only labeled instances from S and test on T. While previous works are restricted to datasets involving only binary classification [5, 6], we also report results on public datasets involving more realistic and useful multi-label multi-class classification setting. Additionally, to verify the effectiveness of our approach in limited labeled data scenarios we reduce the source domain training data. We consider 50%, 25% and 10% of original training set without changing the validation and test sets.

A. Datasets

On datasets involving binary classification, we conduct experiments on two publicly available datasets of tweets; ‘2015 Nepal earthquake’ (NEQ) and ‘2013 Queensland flood’ (QFL) [6]. The associated class information indicates if a tweet is relevant to the disaster, and the task is to predict the (ir)relevance label for tweets in the test set. Table II presents the details of NEQ and QFL datasets. The datasets also include unlabeled tweets, mined in the same time-frame of the disaster, in the form of tweet ids (for exact details, refer to [6]). We used Twitter’s public API to download these tweets. The process resulted in 49, 223 and 15, 464 unlabeled tweets related to NEQ and QFL datasets, respectively. We used the same train, dev and test split as provided by the authors for our experiments.

On datasets involving multi-class multi-label classification, we conduct experiments on ‘Forum for Information Retrieval Evaluation 2016’ (FIRE16) [27] and ‘Social Media for Emergency Relief and Preparedness’ (SMERP17) [28] containing tweets posted during 2015 earthquake in Nepal and 2016 earthquake in central Italy respectively. These datasets contain tweets with granular, actionable class information such as resource available, resource required, infrastructure damage, and rescue activities. FIRE16 contains 7 granular class information, whereas SMERP17 contains 4 classes that are closely related to the FIRE16 classes. Due to lack of publicly available datasets with granular actionable information, we pair up FIRE16 and SMERP17 by mapping the 7 classes of FIRE16 into 4 classes of SMERP17. The grouping is quite natural, and the details are presented in Table III. Once again, these datasets also include unlabeled tweets, and we followed a similar process as for binary datasets to collect tweets. This process resulted in 68, 964 unlabeled tweets for SMERP17. Details of the FIRE16 and SMERP17 datasets after mapping are provided in Table IV and V. Note that the example counts might differ from original datasets. As Twitter policy does not allow direct sharing, tweets were downloaded before the experiments, and some tweets may not be retrieved if they are deleted or made private.

B. Baselines & Models

We compare our approach against SOTA methods in disaster domain. For binary classification, we compare against the methods named DRCN and DAAT presented in [5] and [6] respectively. These methods, to the best of our knowledge, are the most recent methods with SOTA performance on NEQ and QFL datasets. In addition, we also compare our work against BERT [24]. As BERT outperforms other deep learning based methods (e.g. RNN, CNN) on NEQ-QFL dataset pair, shown by authors Li et al. of DRCN [5] in [29] w.r.t. F1 score, we did not run experiments on these method. On multi-label datasets, owing to lack of any baseline, we compare only against BERT. To the best of our knowledge, we are the first to compare against BERT in disaster domain literature on multi-label domain adaptation setting. We fine-tune all layers of the BERT-base-uncased model using the source domain labeled...
data by appending a linear layer as classification head to the BERT model.

To understand the efficacy of various modelling components of our approach, we report experimental results on the following ablations (all of which include the classification layer with cross-entropy loss over the labelled data as in Section III-C). The last one being the overall model:

- **LEN**: Only Local-Example (Section III-B) Graph component, without the Global-Token Graph i.e. trained on $S_L$ using GAT only.
- **S-GLEN**: The Global-Token Graph constructed from tokens in $S$ alone (i.e. $S_L \cup S_U$), along with Local-Example Graph component. GCN over $S_L \cup S_U$ with GAT over $S_L$.
- **T-GLEN**: The Global-Token Graph constructed from tokens in $S_L \cup T_U$, along with Local-Example Graph component. GCN over $S_L \cup T_U$ with GAT over $S_L$.
- **GLEN**: The Global-Token Graph constructed from tokens in $S_L \cup S_U \cup T_U$ along with Local-Example Graph component, i.e. the overall proposed model. GCN over $S_L \cup S_U \cup T_U$ with GAT over $S_L$.

### C. Training Configuration

Hyperparameters such as learning rate and the number of epochs were tuned with the help of validation data with patience parameter set to 4. As we assumed no labeled data was available from target domain, we use the source domain validation set for tuning. labeled data from target domain was used during testing only. Learning rate search space was limited to $\{10^{-3}, 10^{-4}, 10^{-5}\}$ values only. We trained BERT on a system with a ‘Nvidia Tesla P100’ GPU with 12GB GPU-RAM, 96GB system RAM and 56 cores. Our model GLEN was trained on CPU only with 32GB RAM and 20 cores. We used 100 as hidden dimension size for all experiments and 300 dimensional GloVe embeddings were used as initial input token-level features. The number of parameters for GLEN was $\approx 664k$ which is $\approx 165$ times smaller than BERT in terms of number of parameters. All results reported for GLEN and BERT are averaged across 3-runs, each corresponding to a different random seed for the experiment while training. Below we present the architecture details of our model GLEN.

```python
GLEN(
    (token_gcn): GCN(
        (gc1): GraphConvolutionLayer (300 -> 100)
        (gc2): GraphConvolutionLayer (100 -> 100)
    )
    (instance_gat_dgl): Instance_GAT_dgl(
        (conv1): GATConv(
            (fc): Linear(in_features=300, out_features=200, bias=False)
            (attn_drop): Dropout(p=0.0, inplace=False)
            (feat_drop): Dropout(p=0.0, inplace=False)
            (leaky_relu): LeakyReLU(neg_slope=0.2)
        )
        (conv2): GATConv(
            (fc): Linear(in_features=200, out_features=100, bias=False)
            (attn_drop): Dropout(p=0.0, inplace=False)
            (feat_drop): Dropout(p=0.0, inplace=False)
            (leaky_relu): LeakyReLU(neg_slope=0.2)
        )
    )
    (bilstm_classifier): BiLSTM_Classifier(
        (lstm): LSTM(200, 100, num_layers=2, batch_first=True, dropout=0.2, bidirectional=True)
        (fc): Linear(in_features=200, out_features=4, bias=True)
    )
)
```

### D. Results

We report results on three different experiments. Initially we compare the performance of our model with baselines when all source domain training data was available. Then we verify the effectiveness of our approach in low-labeled data regime, i.e. limited training data. We also performed ablation study over different global embedding settings. The results are shown in Table VI and VII.
Finally, we performed qualitative investigation on the capabilities of our model.

1) Results on all available data

Table VII presents a comparison of GLEN with other SOTA methods (DAAT and DRCN) and BERT on the test set of the NEQ and QFL datasets. We observe that for cross-domain setting of NEQ-QFL dataset pair, GLEN outperforms both SOTA methods and BERT by 2.74% and 3.55% on average, respectively, w.r.t. weighted F1 metric. In addition, we observe that GLEN outperforms other methods in same-domain (last two rows of Table VII) by 0.95%. The larger gain in cross-domain compared to same-domain setting indicates that our approach is better suited to domain adaptive setting, which is our original objective. Table VII presents a comparison of GLEN with BERT on the test set of FIRE16 and SMERP17 dataset pair. Since we are the first to investigate UDA in the cross-domain multi-label setting, there are no baselines available and we compare against BERT. We observe that GLEN outperforms BERT in all settings on average 3.00% w.r.t. weighted F1 metric. Once again, GLEN is better than BERT when we train and test on the same domain, showing the robust performance of the proposed approach in general.

2) Results on limited training data

Table VIII summarises the performance of our approach compared to BERT when limited labeled data is available. We restrict the source domain training data to 100%, 50%, 25% and 10% of original training data size. We observed that as the amount of labeled training data decrease in source domain, the performance gain of GLEN over BERT increases as presented in Figure 3. When the models are trained with 10% training data i.e. \( \approx 650 \) examples for NEQ-QFL and \( \approx 100 \) examples for FIRE16-SMERP17 dataset pairs, the performance gain is as much as 3.26% and 17.02% respectively. The results indicate that GLEN can retain performance by utilizing the global token graph effectively. Between binary (NEQ-QFL) and multi-class (FIRE16, SMERP17) setups, the required annotation effort is more for the multi-label multi-class setup. Moreover, with the same effort, the number of examples available per-class is lesser in case of multi-label annotation. We see that the performance difference between GLEN and BERT is more in such scenarios. These findings indicate that GLEN is better in learning from smaller amount of labeled data, which is crucial for disaster management.

3) Component-wise results

In Table IX we present an ablation over various components of the proposed model GLEN. We note that the introduction of global-token graph improves the performance of the model significantly as every model configuration performs better than LEN, which only uses instance (e.g. tweet) level information. In the model configurations, which include the global-token graph, GLEN is better than both S-GLEN and T-GLEN indicating that constructing a graph with tokens from both domains and propagating information is helpful. Performance of T-GLEN is worse in comparison to S-GLEN for every dataset combination; this is possibly due to low vocabulary overlap between labeled source and unlabeled target tokens. Resulting in sparser graph and hence the global-token graph representations can not capture much additional information over original Glove embeddings.

E. Qualitative Investigation

In Section III-A, we claimed that GCN on the global-token graph should bring tokens which appear in similar contexts closer to one another in embedding space. We argued that this is beneficial to mitigate the covariate shift between the domains. We investigate if this holds in practice by designing an experimental study. We selected a small subset of tokens from vocabulary and collected their representation before and after training the network. The token subset is chosen on the basis of a heuristic where some token pairs appear in similar contexts from instances in the datasets. In contrast, some others do not appear in similar contexts. The particular heuristic helps investigate if the GCN component of the model is able to learn domain agnostic representation for tokens. We then calculated the cosine similarity of a token to all the other tokens in the subset using the input embedding (in case of “before”) and output of global token graph (GCN outputs in case of “after”). The cosine similarity values are z-score normalized to have 0-mean and 1-std. deviation. The normalization was performed to investigate the relative similarity.

Figure 2 shows the heatmap of the normalized cosine similarity between pairs of tokens in the selected token subset. Figure 2a is cosine similarity between the initial input (Glove) representation of a pair of tokens. At the same time, Figure 2b shows the same for the token representations from the global-token graph (GCN component) after end-to-end training of the model. We note that the global-token graph
TABLE VIII: Weighted F$_1$ scores for NEQ-QFL and FIRE16-SMERP17 dataset pairs for same and cross domain setting in limited training data regimes. It can be observed that performance gap between our method GLEN and BERT increases with decrease in training data. Average row signifies average performance gain on all four datasets combined.

| Source | Target | BERT | GLEN | Gain % | BERT | GLEN | Gain % | BERT | GLEN | Gain % |
|--------|--------|------|------|--------|------|------|--------|------|------|--------|
| NEQ    | QFL    | 80.49| 83.36| 2.87   | 79.45| 82.52| 3.07   | 78.03| 81.86| 3.82   |
| QFL    | NEQ    | 65.78| 70.33| 4.55   | 65.30| 69.86| 4.56   | 64.62| 69.40| 4.78   |
| NEQ    | NEQ    | 74.07| 75.82| 1.75   | 71.77| 75.01| 3.24   | 71.36| 74.19| 2.83   |
| QFL    | QFL    | 94.73| 96.23| 1.59   | 94.53| 96.14| 1.60   | 94.09| 95.72| 1.63   |

FIRE16 | SMERP17| 72.64| 76.77| 4.13   | 63.05| 74.67| 11.61  | 27.42| 58.23| 30.81  |
SMERP17| FIRE16 | 37.73| 47.33| 9.59   | 28.21| 43.24| 15.03  | 16.71| 33.81| 17.10  |
FIRE16 | FIRE16 | 70.44| 78.68| 8.23   | 58.99| 72.25| 13.26  | 42.73| 57.08| 14.35  |
SMERP17| SMERP17| 91.24| 93.49| 2.24   | 84.03| 88.02| 3.99   | 71.29| 77.14| 5.84   |

Average | 4.37 | 7.05 | 10.15 |

Fig. 3: Average absolute performance gain (weighted F$_1$) of GLEN over BERT as training data is reduced on NEQ-QFL (left) and FIRE16-SMERP17 (right) dataset pairs respectively.

TABLE IX: Ablation with weighted F$_1$ scores for various cross-domain settings. Every GLEN model configuration is better than LEN, S-GLEN and T-GLEN.

| Source | Target | LEN | S-GLEN | T-GLEN | GLEN |
|--------|--------|-----|--------|--------|------|
| NEQ    | QFL    | 81.94| 82.58  | 81.97  | 83.42|
| QFL    | NEQ    | 64.75| 65.44  | 65.13  | 71.61|
| FIRE16 | SMERP17| 74.04| 75.02  | 74.33  | 80.80|
| SMERP17| FIRE16 | 52.41| 56.13  | 55.83  | 56.56|

TABLE X: Normalized cosine similarity values for token pairs from Figure 2 for Glove and GLEN’s Global-Token Graph representations. The Type column indicates whether the tokens should come closer (or move away) in embedding space, basis how they appear in common context.
V. Conclusion

In this paper, we addressed the problem of unsupervised domain adaptation in the area of disaster management on online social networks. To this effect, we proposed a novel two-part graph-based framework which utilizes both labeled and unlabeled data from the source and unlabeled data from the target domain. Our model learns domain agnostic features for tokens using graph convolution networks on token graphs built from labeled and unlabeled data across domains and also preserves local instance level context using a graph attention network on instance graphs. Through our experiments, we showed significant improvements over SOTA methods on 4 datasets involving binary and multi-label classification. We also implement a BERT based baseline for the first time in this area and show that our method outperforms BERT by a significant margin on average, across datasets. We also showed that our model can retain performance in limited data regime due to effective utilization of unlabeled data. We qualitatively investigated the effects of our model and found GLEN brings features of source and target domains closer. In future, we would investigate the performance of our model from domains other than disaster management.
