Multi-task Learning to Enable Location Mention Identification in the Early Hours of a Crisis Event

Sarthak Khanal  
Kansas State University  
sarthakk@ksu.edu

Doina Caragea  
Kansas State University  
dcaragea@ksu.edu

Abstract

Training a robust and reliable deep learning model requires a large amount of data. In the crisis domain, building deep learning models to identify actionable information from the huge influx of data posted by eyewitnesses of crisis events on social media, in a time-critical manner, is central for fast response and relief operations. However, building a large, annotated dataset to train deep learning models is not always feasible in a crisis situation. In this paper, we investigate a multi-task learning approach to concurrently leverage available annotated data for several related tasks from the crisis domain to improve the performance on a main task with limited annotated data. Specifically, we focus on using multi-task learning to improve the performance on the task of identifying location mentions in crisis tweets.

1 Introduction

Social media has evolved into a platform for people to share their concerns, report information as eyewitnesses of events, and also call for help, especially during crisis situations. The huge amount of data that is posted on social media during crisis events could be used to build reliable and robust deep learning models for identifying information useful to crisis management and response teams. However, using social media data for a particular task, oftentimes, requires intensive manual effort in the form of annotation. The effort becomes even more arduous when we consider the noisy nature of social media content and the amount of labelled data required for a typical deep learning model.

The domain of crisis-related social media analysis, tweets in particular, is a well-researched field with labelled data available for various tasks (Imran et al., 2016; Middleton et al., 2014; Alam et al., 2018). However, most of the available human-annotated datasets consists of thousands of instances, at best, which means that crisis datasets are relatively small compared to those available for tasks in other domains. Furthermore, for tasks that can support a new, emergent crisis situation, human-labelled data of large volume cannot be acquired for the reasons discussed above. In this work, we explore ways in which we can harness the available small datasets from the domain to bolster performance for individual tasks of interest.

One popular approach in addressing the size-limitation of labelled data for a particular task is to leverage unlabelled data. In the field of Natural Language Processing (NLP), the recent advancements in transformer-based architectures, and associated pre-training with huge amounts of unlabelled data, has largely been successful in addressing this issue. Transformer-based architectures, currently, hold state-of-the-art results for many NLP tasks. However, the domain shift from the pre-training corpus to a downstream task’s domain is still a significant issue (Han and Eisenstein, 2019). Moreover, further pre-training with domain-specific unlabelled data is compute-intensive (Devlin et al., 2018) and it is not always feasible.

An alternative approach to address the limitation in terms of labeled data is to concurrently leverage smaller datasets available for different, but related tasks using a multi-task learning strategy (Caruana, 1997). In the multi-task setting, some layers can be shared across different tasks, while each task can also have one or more task-specific layers, and the entire model is trained in parallel for all the tasks. A multi-task model is designed with the intuition that the lower layers of the model learn abstract features common to related tasks, while the upper layers learn features specific to each individual task. This approach is especially useful in the domain of crisis-related social media analysis, given the lack of large datasets, while smaller datasets are available for different tasks.

In this work, our main focus is on the task of identifying fine-grained locations from tweet texts.
Identification of location entities in tweets posted during crisis events is vital in extracting actionable situational awareness information. Furthermore, identifying the entities according to a hierarchy of location types can help in geographical location disambiguation and geo-coding. We use the English subset of the dataset published by Middleton et al. (2014), which has only a few thousands instances.

To address the issue of limited data size, we use a multi-task learning setting to augment the learning of fine-grained location identification with three other tasks in the domain of crisis-related tweets: key-phrase identification (Chowdhury et al., 2020), eyewitness-account classification (Zahra et al., 2020) and humanitarian categories classification (Alam et al., 2018). We hypothesize that the similar nature of the tasks and the common abstract objective of identifying actionable information in crisis-related tweets will result in a performance boost for the main task considered, i.e., identification of fine-grain locations in tweet texts.

2 Related Work

Multi-task learning is a well-researched topic in the field of NLP, and deep learning, in general (Ruder, 2017). Caruana (1997) outlines one of the popular strategies for implementing multi-task learning: hard parameter sharing. In hard parameter sharing, a module shared by all tasks is followed by task-specific modules. On the other hand, in soft parameter sharing (Duong et al., 2015; Yang and Hospedales, 2017), each task has its own set of layers, but the parameters of the task-specific layers are constrained to be similar across tasks to enforce exchange of information among tasks. Hard parameter sharing is known to reduce over-fitting (Ruder, 2017), and is thus useful for tasks with small training sets. The usefulness of multi-task learning has been shown in a variety of applications, including image (Zhang et al., 2014; Cheng et al., 2011), voice (Stoller et al., 2018; Rao et al., 2018) and text (Wang et al., 2020; Liu et al., 2019a; Pham et al., 2019) analysis. In the crisis domain, Wang et al. (2020) presented a multi-modal multi-task model using a single multi-modal dataset containing labels for different tasks. Chowdhury et al. (2020) used single-token keywords identification as an auxiliary task when predicting multi-token keyphrases. The work by Liu et al. (2019a) is the closest to our work, in that they used separate task-specific datasets in a multi-task learning setting. However, they performed experiments on general NLP benchmark datasets, such as GLUE (Wang et al., 2018).

3 Background and Approaches

We use the hard-parameter sharing approach for multi-task learning, where task-specific modules ($τ$) are attached on top of a shared module ($γ$) as shown in Figure 1. We use the algorithm proposed by Liu et al. (2019a) to train multiple task modules, in parallel.

**Multi-task Learning.** Multi-task learning can be formulated as follows: Given a set of $k$ tasks and their corresponding data $D = \{(X_1,Y_1),(X_2,Y_2),\cdots,(X_k,Y_k)\}$, where $(X_i,Y_i) \in D$ is the training dataset for the $i^{th}$ task, the goal of multi-task learning is to minimize the aggregate loss $L(\theta)$ on $D$ given by:

$$L(\theta) = \sum_{i=1}^{k} \sum_{(x,y) \in (X_i, Y_i)} \ell^i(y, T^i(x, \theta))$$  \hspace{1cm} (1)

where $\ell^i$ is the loss function for the $i^{th}$ task, $T^i(x, \theta) = \tau^i(\gamma(x, \theta_\gamma), \theta_i)$ is the output of the $i^{th}$ task module $\tau^i$ (with parameter $\theta_i \subset \theta$) and $\gamma$ is the shared module (with parameter $\theta_\gamma \subset \theta$).

**Token Classification Tasks.** A token classification task maps a sequence $x = \{x_1, \cdots, x_n\}$, representing $n$ input tokens, to a sequence $y = \{y_1, \cdots, y_n\}$, representing $n$ task-specific tags corresponding to the tokens in the sequence $x$. The mapping can be generically written as: $\tau^i(\theta_i) : x \rightarrow y$ (where $\theta_i$ are the task parameters). The loss $\ell^i$ that we use for these tasks is the negative log likelihood. In our setting, the fine-grained location identification and the keyphrase identification tasks are token classification tasks.
Sequence Classification Tasks. A sequence classification task maps a sequence of $n$ tokens $x = \{x_1, \ldots, x_n\}$ to a class $y \in C$, where $C$ is the task-specific set of classes. The mapping can be generically written as $\tau^j(\theta_j) : x \rightarrow y$ (where $\theta_j$ are the task parameters). We use the standard cross-entropy or the binary cross-entropy loss $\ell^j$ for sequence classification tasks, depending upon the number of classes. In our setting, the eyewitness-account classification and the humanitarian categories classification tasks are seen as sequence classification tasks.

Shared Module. For the shared module, we experiment with three base variants (hidden layer size of 768) of transformer-based language models: BERT (Devlin et al., 2018), Albert (Lan et al., 2019) and RoBERTa (Liu et al., 2019b). BERT is one of the first and most popular transformer-based models, and has been used widely in various applications. We compare its performance with that of RoBERTa, which was trained on a larger dataset than BERT and it is considered to be more robust (Liu et al., 2019b). We also compare BERT and RoBERTa with Albert, a significantly smaller model, to understand the effect that the size of the shared module has on the performance of the task at hand.

4 Experimental Setup

4.1 Datasets

Fine-grained Location Identification. We use the dataset published by Middleton et al. (2014), which has two sets of tweets posted during Hurricane Sandy 2012 and Christchurch Earthquake 2012, respectively. Based on the place of the event, in what follows, the set of tweets posted during Hurricane Sandy 2012 will be referred to as NY, while those posted during Christchurch Earthquake 2012 will be referred to as NZ. The NY and NZ sets contain 1907 and 1762 unique tweets, respectively. Both the NY and NZ tweets have human-labelled location entities corresponding to three categories: administrative location, building and transportation. We randomly select 1000 tweets from each set for training, 500 tweets for test, and the remaining tweets for development (dev) splits.

Keyphrase Identification. We use the dataset published by Chowdhury et al. (2020) for keyphrase identification. The dataset contains tweets from various crisis events. The tokens in each tweet are labelled as keyphrase or not, using the script provided by Chowdhury et al. (2020). We use a random sample of 1000 tweets for training to keep the dataset size balanced across multiple tasks.

Eyewitness-account Classification. The dataset published by Zahra et al. (2020) contains tweets from flood, hurricane and earthquake. The tweets are labelled using one of five eyewitness classes: direct-eyewitness, indirect-eyewitness, vulnerable direct-eyewitness, non-eyewitness, and don’t know. As for the other tasks, we use a class-balanced sample of 1000 tweets from the dataset for training.

Humanitarian Categories Classification. We use the dataset published by Alam et al. (2018) for humanitarian categories classification. The dataset contains tweets from various crisis events, labelled using the following humanitarian classes: Infrastructure and utility damage, Vehicle damage, Rescue, volunteering, or donation effort, Injured or dead people, Affected individuals, Missing or found people, Other relevant information, and Not relevant or can’t judge. We use a class-balanced sample of 1000 tweets from the dataset for training.

4.2 Baseline and Metrics

We use a single task-setting, where only one task-module is attached over the base module, as the baseline to which we compare the performance of the multi-task learning setting. We use precision (Pr), recall (Re) and F1 scores as metrics to compare the performance of the models.

4.3 Experiments and Implementation Details

We perform the experiments on three levels of task-sharing: a Single-task model (ST) (trained on NY and NZ, respectively, and tuned/tested on the corresponding dev/test sets), a Location-only Multi-task model (LMT) (trained with NY and NZ as two different location identification tasks in a multi-task setting, and tuned/tested on the corresponding dev/test sets), and Multi-task model (MT) (trained with all tasks in a multi-task setting, and tuned/tested on the corresponding dev/test sets for the location identification tasks). In addition, we also combine the two location datasets NY and NZ (Combined) and train a single-task model (ST), and a multi-task model (MT), using the other three tasks to understand the benefit of only using the related tasks for general location identification (as opposed to disaster-specific identification). The hyperparameter configuration used in the experiments are shown in the Appendix A. All experiments are run on a 4-GPU system of NVIDIA Tesla V100 GPUs.
Table 1: Results for Fine-grained location identification task in three settings: Single-task (ST), Location-only Multi-task (LMT) and all-multi-task (MT) with three variants of transformers (γ) as shared module. The results are grouped by dataset: NY, NZ and Combined.

<table>
<thead>
<tr>
<th>Task</th>
<th>Dataset</th>
<th>ST</th>
<th>LMT</th>
<th>MT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BERT</td>
<td>75.40</td>
<td>75.49</td>
<td>73.95</td>
</tr>
<tr>
<td></td>
<td>RoBERTa</td>
<td>66.31</td>
<td>69.09</td>
<td>70.12</td>
</tr>
<tr>
<td></td>
<td>Albert</td>
<td>72.40</td>
<td>76.09</td>
<td>77.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70.18</td>
<td>82.35</td>
<td>84.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79.41</td>
<td>79.10</td>
<td>78.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79.30</td>
<td>79.24</td>
<td>78.75</td>
</tr>
</tbody>
</table>

Table 2: Average (macro) performance improvement over three datasets for the three transformer variants, when comparing single-task model (ST) to location-only multi-task (LMT) and all-multi-task (MT) model.

<table>
<thead>
<tr>
<th>Transformer</th>
<th>Average performance improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>ST → LMT -0.78 ST → MT 3.43</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>4.75 2.36</td>
</tr>
<tr>
<td>Albert</td>
<td>4.17 6.35</td>
</tr>
</tbody>
</table>

6 Conclusions and Future Work

In this paper, we studied the effect of using other closely related tasks in a multi-task setting for the task of fine-grained location identification in the domain of crisis-related tweets. Our results show that using multi-task learning can improve the performance significantly as compared to single-task learning. The approach could be specially useful when analysing an emergent crisis, where annotating large amount of data is not feasible and the entire analysis process is time-critical. Moreover, our results also show that when a small amount of data is available, the model may benefit more from using other related tasks rather than using additional data for the same type of task (i.e., two location identification tasks).

In addition to studying the effect of multi-task learning, we also studied the effect of different transformer variants on the performance of the model for the main task. With different variants available, it is necessary to find the best-fitting one for the task at hand. The results show that the choice of the shared module has significant effect on the performance.
As part of future work, we plan to explore other multi-tasking and domain adaptation strategies with different collections of datasets to allow for stress testing of the approach in terms of generalization. Given the promising results with multi-task learning, we plan to perform studies with respect to the nature of the crisis for the dataset. This will allow us to focus on specific datasets for specific crises and potentially improve the state-of-the-art on those datasets.

Acknowledgements

We thank the National Science Foundation and Amazon Web Services for support from grant IIS-1741345, which supported the research and the computation in this study.

References


A Hyperparameter configuration

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning rate</td>
<td>1e-5</td>
</tr>
<tr>
<td>Learning rate decay</td>
<td>Linear</td>
</tr>
<tr>
<td>Batch size</td>
<td>{4, 8, 16}</td>
</tr>
<tr>
<td>Classifier Dropout</td>
<td>0.1</td>
</tr>
<tr>
<td>Optimizer</td>
<td>AdamW</td>
</tr>
<tr>
<td>Adam epsilon</td>
<td>1e-8</td>
</tr>
<tr>
<td>Maximum Epochs</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 3: Hyperparameter configuration used in the experiments. The best performing setting is highlighted in bold face.