2021 IEEE 17th International Conference on eScience (eScience) | 978-1-6654-0361-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/eScience51609.2021.00052

CrisisFlow: Multimodal Representation Learning Workflow for Crisis Computing

Patrycja Krawczuk
Department of Computer Science
University of Southern California
Los Angeles, USA
krawczuk@usc.edu

Shubham Nagarkar
Department of Computer Science
University of Southern California
Los Angeles, USA
slnagark@usc.edu

Ewa Deelman
Department of Computer Science
University of Southern California
Los Angeles, USA
deelman@isi.edu

***Abstract*—An increasing number of people use social media (SM) platforms like Twitter and Instagram to report critical emergencies or disaster events. Multimodal data shared on these platforms often contain useful information about the scale of the event, victims, and infrastructure damage. The data can provide local authorities and humanitarian organizations with a big-picture understanding of the emergency (*situational awareness*). Moreover, it can be used to effectively and timely plan relief responses. In our project, we aim to address the challenge of finding relevant information among the vast amount of published SM posts. Specifically, we use deep learning algorithms to produce embeddings that encode the *informativeness* of multimodal SM data in the context of disaster events. Our method improves the state-of-the-art performance on the *informative vs. non-informative* classification task for the CrisisMMD dataset. To ensure the reliability and scalability of our solution in real-world scenarios, we implement the resulting crisis computing workflow in the Pegasus Workflow Management System (WMS).**

I. Introduction

Global warming exacerbates the frequency and scale of extreme weather events such as floods, hurricanes, wildfires, cyclones, and blizzards. Natural disasters require quick and targeted emergency responses to save human lives and to mitigate infrastructure damage. Increasingly, SM platforms serve as indispensable tools to acquire data needed to locate victims and plan efficient emergency reliefs (see Fig.1), especially in the early hours after a disaster's onset.

Fig. 1. There are many real-world examples of the use of Twitter data to save lives during crisis events. The news articles report use of SM for targeted emergency responses during the 2011 Tohoku earthquake and tsunami in Japan, and the 2018 Hurricane Harvey in Huston [1].

One of the biggest obstacles in the use of SM content for crisis response is handling the information overload. As millions of SM posts are published at any time, there is a need to develop a system capable of processing a massive amount of data in near real-time. Such a system should be able to filter through thousands of short, informal messages and large amounts of images to extract the SM posts that are credible and build situational awareness during a disaster event.

In this project, we aim to improve the accuracy of the existing deep learning methods on *informative vs. non-informative* classification task on the CrisisMMD [2] dataset. We develop CrisisFlow, a computational workflow capable of processing and classifying thousands of multimodal SM posts.

II. Dataset

The CrisisMMD dataset is the most widely used benchmark for multimodal learning for crisis computing [3]. It consists of thousands of annotated image-text pairs collected during seven major natural disasters such as earthquakes, hurricanes, wildfires, and floods. It includes three types of annotations: *informative vs. non-informative, humanitarian categories* and *damage severity assessment*.

III. Methodology

A. Contrastive Learning on Images

Contrastive learning methods learn embeddings by pulling the representations of similar images together and pushing dissimilar ones apart from each other in the latent space. The methodology is often applied to problems where labeled data is not available. In this project, we utilize the SupCon model introduced in [4]. This semi-supervised architecture leverages the labeled data through a novel *Supervised Contrastive Loss*. The loss function is designed to increase its discriminatory power when provided with a large number of negative examples (in our case *non-informative* pictures).

B. Sentence Embeddings

Sentence embeddings encode the semantics and structure of a sentence into a vector. The embeddings are often generated by extracting several hidden layers from a language model and averaging them. To create task-relevant sentence representations, researchers often use a pre-trained language model that is fine-tuned to a given classification task. In this project, we fine-tuned the DistilBERT [5] model to create objective-specific embeddings for our downstream classification task.

978-1-6654-0361-0/21/$31.00 ©2021 IEEE
DOI 10.1109/eScience51609.2021.00052

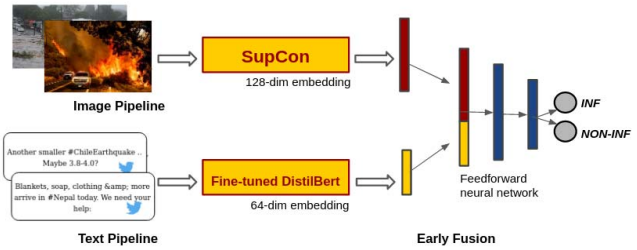

Fig. 2. Our architecture consists of two pipelines that ingest pictorial and textual parts of SM posts, respectively. The vector representations produced by the pipelines are then concatenated and fed into a neural network that classifies the post as either *informative* or *non-informative*.

C. Multimodal Learning: Early Fusion

Early fusion is a technique that allows merging data from different sources and modalities (image, text, audio) at a vector level. The multiple modalities often offer complementary information that helps to improve the performance of a model. Our architecture consists of two pipelines that ingest pictorial and textual parts of SM posts, respectively. The modalities serve as inputs to the architectures that generate the crisis-specific representations of the data as low-dimensional vectors. The image and text embeddings are concatenated and passed through a neural network that classifies the SM post as *informative* or *non-informative*.

IV. Experimental Results

a) Images: To produce meaningful, low-dimensional representations of the data, we train the SupCon model for 150 epochs with a batch size of 4, SGD-Momentum optimizer, and 0.5 as the mean and standard deviation value to normalize the images. After the training, we remove the classification head and use the intermediate trained layers to generate the embeddings. We evaluate the quality of the representations with a kNN method. Although training SupCon is computationally expensive and needs a minimum of 700 epochs, we can achieve an accuracy of about 76% using a single GPU.

b) Text: The DistilBERT is pre-trained on the GloVe Twitter 27 B embeddings. We fine-tune the model for 4 epochs with a batch size of 16. We use the Adam optimizer with a weight decay of 0.01 and a custom weighted loss function that compensates for the unbalanced dataset. We extract the sentence embeddings from the fine-tuned model by averaging all the final hidden layers of all the tokens in the sentences. We generate the latent representations for both the training data and test data that the model has not seen before.

c) Images and Text: The image embeddings of size *1x128* and their corresponding text representations of size *1x64* are merged into a single vector. The vector serves as an input to a simple feed-forward network with a softmax as its last layer (see Fig.2). We train the network for 30 epochs with a learning rate of 0.001, batch size of 32, and Adam optimizer. We compare the results of our experiments with a baseline multimodal architecture designed by Olfi et al. [3]. CrisisFlow successfully outperforms the baseline, as seen in Table 1.

TABLE I
Comparison of the results between *Olfi et al.* and CrisisFlow .

Data Modality	Method	
	Olfi et al.	*CrisisFlow*
Text	0.808	**0.84**
Image	0.833	**0.89**
Image + Text	0.844	**0.91**

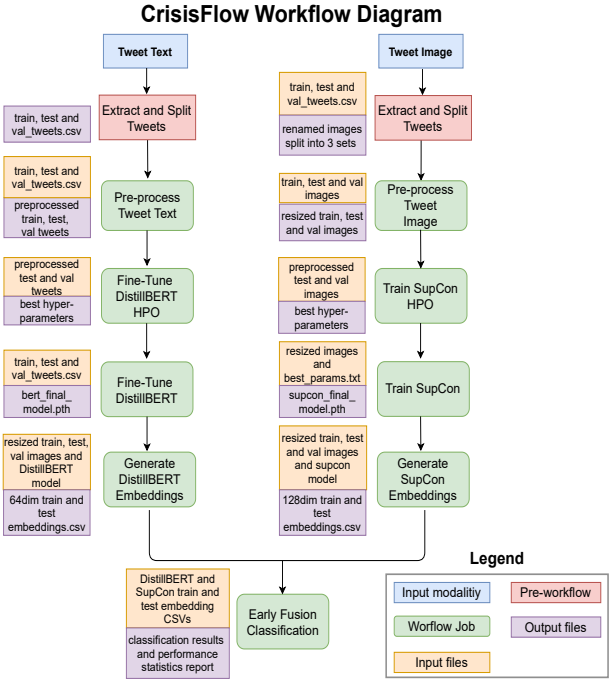

Fig. 3. The CrisisFlow graph describes job dependencies, input and output files in the workflow. The CrisisFlow is implemented in Pegasus WMS.

d) CrisisFlow: To ensure the reliability and scalability of our solution in real-world scenarios, we implement our method, CrisisFlow, in Pegasus WMS [6]. The directed acyclic graph (DAG) of the CrisisFlow where the nodes represent individual tasks while edges represent dependencies between them is depicted in Fig. 3. Pegasus automatically maps the DAG onto available distributed resources. Tasks like pre-processing and hyper-parameter tuning can be executed in parallel. Researchers can use CrisisFlow to train the binary classification models on their own SM data. The workflow expects as inputs a directory of images and a csv file with tweets that contains: *tweet's id*, *tweet's text*, *tweet's label*, *ids of associated images*. The csv file and the images are added to Pegasus' *Replica Catalog*.

V. Conclusion and Future Work

The use of SM content presents an opportunity to provide timely and actionable information to first responders and local

officials during a disaster event. However, there are many challenges before we can take full advantage of the data to build *situational awareness* during a crisis. One of them is the issue of information overload. In this work, we implement and train CrisisFlow, a scalable workflow that can process large amounts of multimodal SM data and classify them as *informative* or *non-informative*. In the future, we aim to generate a coherent summary of a disaster event and further improve the performance of our method.

VI. ACKNOWLEDGEMENTS

This work is funded by NSF contract 1664162, "SI2-SSI: Pegasus: Automating Compute and Data Intensive Science".

REFERENCES

[1] M. Koren, "Using Twitter To Save A Newborn From A Flood". The Atlantic, 2017

[2] F. Alam, F. Ofli, M. Imran, "CrisisMMD: Multimodal Twitter Datasets from Natural Disasters". Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM), 2018

[3] F. Ofli, F. Alam, M. Imran, "Analysis of Social Media Data using Multimodal Deep Learning for Disaster Response". 17th International Conference on Information Systems for Crisis Response and Management (ISCRAM), 2020

[4] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, "Supervised Contrastive Learning". Conference on Neural Information Processing Systems (NeurIPS), 2020

[5] V. Sanh, L. Debut, J. Chaumond, T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter", 2020

[6] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. Maechling, P. Mayani, W. Chen, R. Silva, M. Livny, K. Wenger, "Pegasus: a Workflow Management System for Science Automation". Future Generation Computer Systems, 2015, pp. 17-35