Multimodal Learning for Disaster Assessment Using Social Media Data

Swetha Revanur
Stanford University
srevanur@stanford.edu

Keanu Spies
Stanford University
keanus@stanford.edu

Abstract

The quantity of social media data is increasing en masse, and it's rapidly becoming intractable to employ human labeling to take advantage of it. Nevertheless, given the immediacy of social media platforms, they can be leveraged as tools for real-time alerts in the case of disasters or wars. Our goal is to automate disaster classification (encompassing natural, human, and infrastructure damage) from social media posts to effectively channel relief resources. Several social media posts are comprised of images and text captions. After exploring a variety of unimodal methods, we propose a federator model that determines whether the text-only or image-only predictions are more likely to be correct. By introducing this model selection step, we enable the unimodal classifiers to complement each other, and as such, achieve state-of-the-art performance at 99% *accuracy. These results highlight the importance of multimodal analyses for disaster assessment.*

1. Introduction

Technology and design innovations are driving the growth of the social media industry. As a consequence, users contribute a diversity of information to the 2.5 quintillion bytes of data created each day [3], largely in the form of images, text, and videos. Social media is particularly useful in disaster response and communication, and can be leveraged as a tool for real-time alerts in the case of natural disasters or wars. However, relief agencies and law enforcement officials struggle to reasonably parse the huge influx of information and assess their confidence in crowdsourced media, given the susceptibility of the platforms to spammers and adversarial attacks. Furthermore, in social media analyses, computationally expensive multimodal approaches are often overlooked in favor of text-only, image-only, or network techniques. Using multimodal social media data can help to counter the effects of false press coverage, rumors, or other reputational issues that plague crisis situations in particular.

Specifically, our goal is to demonstrate that a multimodal approach yields higher damage classification performance compared to text-only or image-only baselines. Our approach will take in text and images, and produce a classification output as detailed in the subsequent section. We evaluate performance with a number of metrics including accuracy, precision, recall, and F1 score.

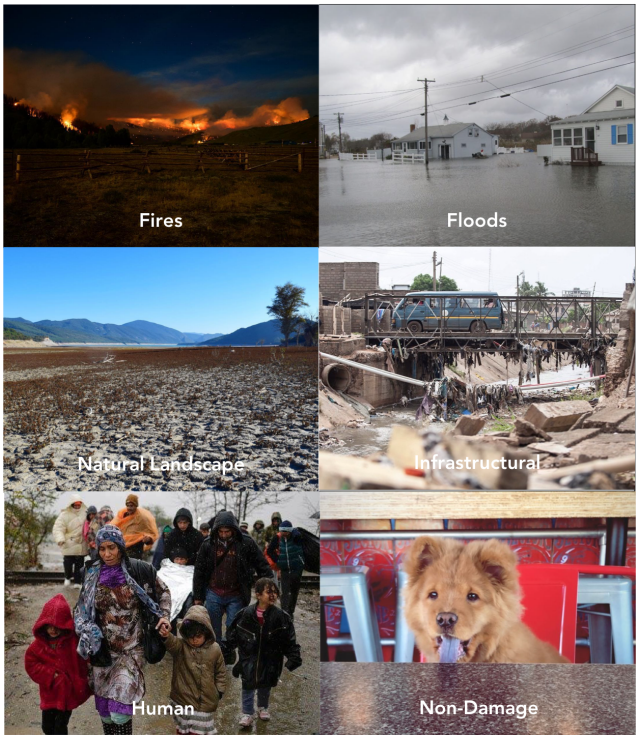


Figure 1. Selected images from each class.

2. Dataset

Here, we seek to extract actionable information from multimodal social media posts to effectively deliver relief resources. We will work with the "Multimodal Damage Identification for Humanitarian Computing" dataset consisting of 5831 captioned images from Twitter, Instagram, and Google Images [4]. Each image and its corresponding caption (jointly referred to as posts) have one of six dam-

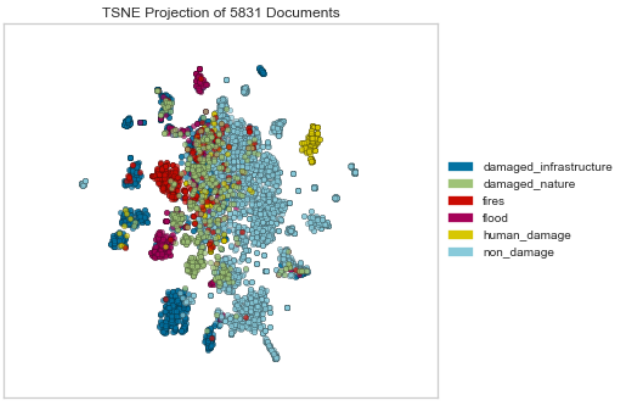


Figure 2. t-SNE of the complete caption dataset with labels.

age labels: fires, floods, natural landscape, infrastructural, human, and non-damage. Example images can be found in Figure 1.

2.1. Data Preprocessing

Each modality requires its own preprocessing steps. For text, we perform casefolding to convert all characters to lowercase. We remove all punctuation including "#" signs. Emojis are retained throughout all steps. Images require different transformations. Given the range of resolutions, images are first resized to be 224×224 then normalized.

2.2. Data Exploration

Since text is often high-dimensional, it's useful to project it onto a lower-dimensional hyperplane for visualization. We apply t-distributed stochastic neighbor embeddings (t-SNE) in two dimensions, which converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data [5]. This t-SNE visualization is shown in Figure 2.

3. Related Works

Ensembling methods are a way of training and combining multiple models with the goal of boosting the accuracy and performance on some problem. Boosting and bagging are the most popular ensembling methods used to date. More complete descriptions can be found in "Ensemble Methods Foundations and Algorithms" by Zhi-Hua Zhou [11].

Zahavy et al. [10] proposed an approach for multimodal product classification using text and image inputs. Namely, they trained two convolutional neural networks (CNNs) for image and text classification separately and used a decision rule to identify which of the two networks to use given a $(text, image)$ tuple. While they did mention exploring combinations of deep image and text features, they did not cite any results and claim it performed sub-optimally compared to other results.

Mouzannar et al. [4] described the problem space of multimodal damage identification from social media posts. They initially approached the problem with unimodal text and image classification techniques, and next proceeded with multimodal approaches to combine the predictions of the two networks. They achieved at best a 92.62% accuracy after fusing two CNNs with decision rules.

Dimitrakakis et al. [2] applied reinforcement learning to ensemble classifiers, which can be used for a multimodal task. The authors framed the environment as n experts implemented as multi-layer perceptrons, and a controlling agent meant to choose between them. A modified, online Q-learning update rule was used to search for an optimal policy. Their experimental results consistently showed $< 5\%$ error rates, although these gains are marginal relative to more computationally-efficient baselines. Motivated by this paper and to exploit the real-time nature of social media, we focus exclusively on methods that are not resource-heavy.

Alqhtani et al. [2] proposed a two-step procedure for detecting events on Twitter. First, textual features were extracted from a bag-of-words model. Then, visual features including histogram of oriented gradients descriptors, grey-level co-occurrence matrix, and color histograms were detected. The final classification decision was based on the relative reliability of the unimodal detections. Since this paper was published, however, deep learning methods have demonstrated higher perceptive capabilities for complex images and text than the simpler features used here.

4. Methods and Experiments

4.1. Unimodal Approaches

To demonstrate the utility of multimodal learning, we establish unimodal baselines for comparison.

4.1.1 Text-Only Approaches

4.1.1.1 Traditional Approaches We begin with text-only approaches. In particular, we explore multinomial naive Bayes, Bernoulli naive Bayes, linear kernel SVMs, and extra-tree models. Multinomial naive Bayes models capture word frequencies, while Bernoulli-based models instead capture word presence and absence. Linear kernel SVMs are used extensively in text classification tasks because text tends to be linearly separable and high-dimensional. In addition, the linear kernel has low computational complexity. Extra-trees (short for extremely randomized trees) use the entire sample at each decision step, while the boundaries are picked at random [6]. We use Word2Vec word embeddings as inputs into these classifica-

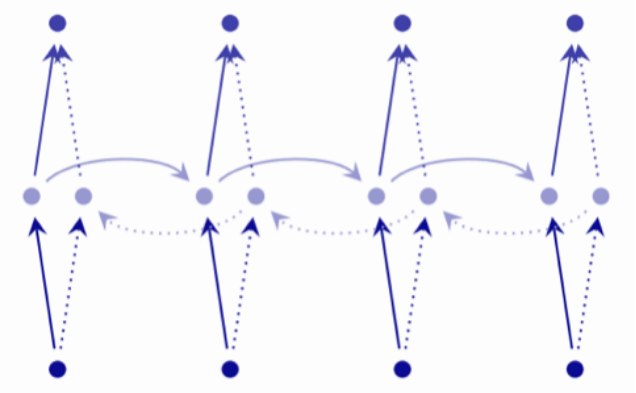
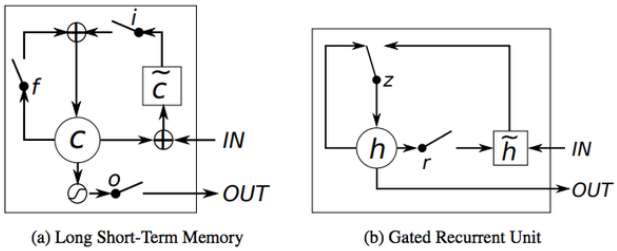


Figure 3. Diagram of a bidirectional RNN, which takes in inputs from the previous and future layers to make contextual decisions at a current layer.



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Figure 4. Diagrams of LSTM and GRU units. The LSTM contains input *i*, forget *f*, and output *o* gates, while GRU has update *z* and reset *r* gates. LSTM has the internal memory state **c** while GRU does not [8].

tion algorithms. Word2Vec learns embeddings using shallow networks and a continuous bag-of-words training architecture. This predicts the current word from a window of neighboring words.

However, Word2Vec embeddings do not take into account word importance. To address this issue, we consider term frequency-inverse document frequency (TF-IDF) variants of the above algorithms, which scale the word vectors by their frequency in the documents. TF-IDF reflects the importance of a given word in a document in a corpus [7].

4.1.1.2 Deep Approaches In an effort to train the most optimal baselines we train multiple RNNs for classification. This process passes the bidirectional RNN output (treated as a deep encoding as shown in Figure 3) through a dense layer to produce a classification prediction.

In order to ensure that the RNNs themselves achieve the optimal performance, we explore LSTM (long short-term memory) and GRU (gate recurrent unit) adaptations to take into account long-term dependencies between words in the text. These variations use gates to regulate the flow of information into and out of the cell as shown in Figure 4.

In addition, we include self-attention to enable longer-term information flow by relating different positions of a single sequence in order to compute a representation of the same sequence [9].

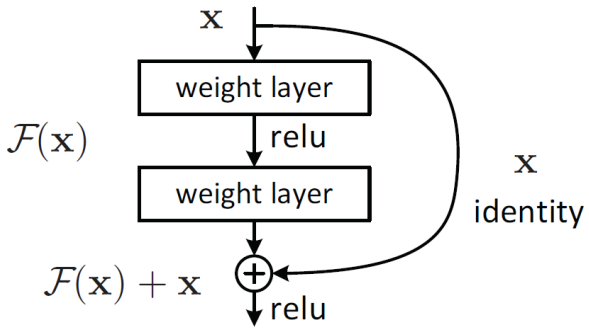


Figure 5. Diagram of a residual block from which ResNets are comprised. The input x is added to the output of the weight layers $F(x)$ [1].

4.1.2 Image-Only Approaches

To train the image-only classifier we take advantage of transfer learning and finetune two pre-trained networks, namely ResNet-18 and ResNet-50, for our classification problem. Residual networks (i.e., ResNets) are deep networks that use residual blocks to create shortcut connections between parts of the network to prevent overfitting. See Figure 5.

Specifically, we finetune the last layer for five epochs and then train the entire network for another five. We use a multi-class cross entropy loss and a decaying learning rate to train. Classification results for all unimodal approaches are outlined in Table 1.

4.2. Multimodal Approaches

4.2.1 Feature Fusion

Feature fusion (FF) combines the internal representations generated from layers of the RNN and CNN to train a multimodal classifier. We extract the second-to-last layer of the two best-performing deep models (the bidirectional LSTM RNN with self-attention and ResNet-50) and concatenate their outputs before passing them through a multimodal neural network. Taking in the combined deep representation of the two models, the system can use compressed knowledge of both text and image data to determine a classification.

To train the multimodal neural network, we use multi-class cross entropy loss with L2-regularization. Using a grid search, we tune hyperparameters including initial learning rate, learning rate decay, dropout rate, batch size, and weight initializations. Early stopping, batch normalization, and regularization are included to reduce variance. We use

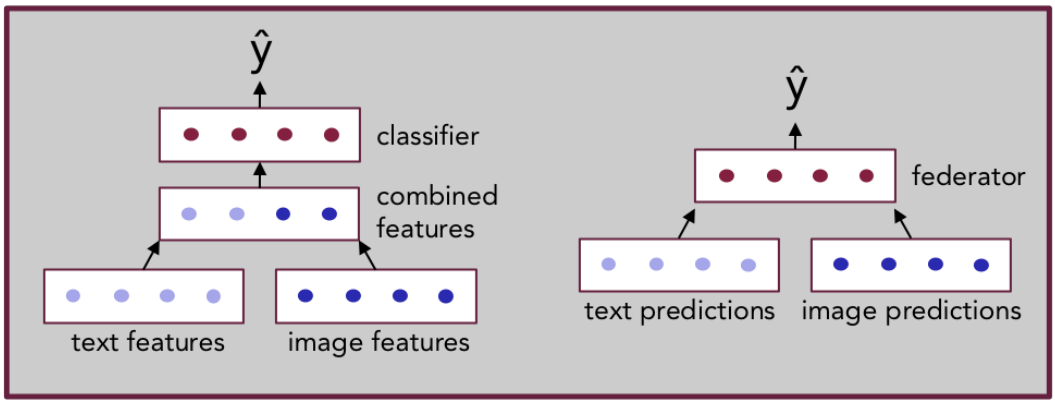


Figure 6. Left: Diagram of feature fusion pipeline.; Right: Diagram of decision fusion pipeline.

stochastic gradient descent with momentum to optimize the objective function. In addition, we varied the number and size of hidden layers and the activation functions, ultimately finding that a single dense layer with batch-normalized inputs works best.

4.2.2 Decision Fusion

In contrast with FF, decision fusion (DF) combines the predictions of the best unimodal classifiers (SVM with TF-IDF and ResNet-50) to obtain better predictions by training a binary classifier. DF is distinct from bagging because there is no notion of sampling smaller training sets, and boosting because there is no reweighting. Instead, we propose an ensembling technique with what we call a federator module. The federator, when given a caption, determines if the caption alone would provide a sufficiently-confident classification decision. If not, its corresponding image is fed through the ResNet-50 to produce the final classification decisions. More concretely, the federator is a classifier that returns 0 if the text predictions should be used, and 1 otherwise.

To train the federator, we generate a modified set of labels. We transform the text and image predictions to binary labels (correct or not). Our final federator labels are positive if the transformed image prediction is strictly greater than the text prediction (i.e., the image-only classifier was correct while the text-only one was incorrect), and negative in all other cases. The federator is trained on the captions only, and as such the federator itself was an SVM with TF-IDF. Since the federator adds additional overhead to the classification process, we opted to train exclusively on captions instead of images because we found that text-only models have substantially lower train and inference times. Specifically, the federator SVM with TF-IDF trained in 6.9583s. At inference time, it took 2.5922s. Classification results for both of the multimodal approaches are outlined in Table 2.

5. Results and Discussion

We see that the linear kernel SVM with TF-IDF-weighted word vectors is the top-performing text-only approach, with an accuracy just above 90%. TF-IDF variants perform better than their usual counterparts for SVMs and extra-trees, but see an approximately 14% decrease in accuracy with multinomial naive Bayes. The top-performing image-only model is the ResNet-50 at around 83% accuracy, slightly higher than ResNet-18.

Model	Precision	Recall	F1-Score	Accuracy
MultiNB	0.8769	0.6940	0.7748	0.8472
MultiNB_TF	0.8882	0.4280	0.5777	0.7098
BernNB	0.4786	0.2845	0.3569	0.6680
BernNB_TF	0.4786	0.2845	0.3569	0.6680
SVM	0.8565	0.8208	0.8383	0.8906
SVM_TF	**0.8942**	**0.8256**	**0.8585**	**0.9052**
Tree	0.5755	0.4687	0.5167	0.6853
Tree_TF	0.6037	0.4872	0.5392	0.6906
LSTM	0.8415	0.7144	0.7516	0.7144
GRU	0.6933	0.6124	0.5231	0.6124
ResNet-18	0.8196	0.8165	0.8161	0.8165
ResNet-50	**0.8352**	**0.8345**	**0.8329**	**0.8345**

Table 1. Results from the Unimodal Techniques

Comparatively, both FF and DF lead to performance gains. In particular, DF is the best of all models at nearly 99% accuracy. Given that the previous highest performance for this dataset and task was 92.6% accuracy, to the best of our knowledge, our DF model has successfully surpassed the state-of-the-art.

Model	Precision	Recall	F1	Accuracy
Feature Fusion	0.9158	0.9151	0.9146	0.9151
Decision Fusion	**0.9890**	**0.9889**	**0.9888**	**0.9889**

Table 2. Results from the Multimodal Techniques

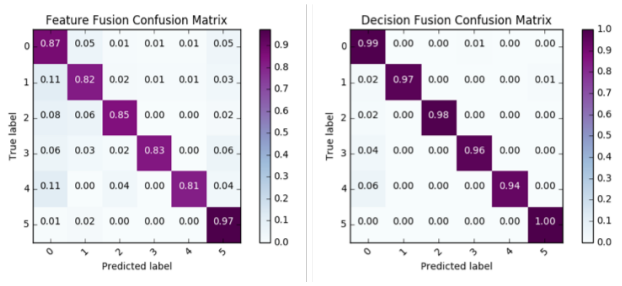


Figure 7. Confusion matrices for fusion methods.

Figure 8. Predictions from the decision fusion.

From the confusion matrices in Figure 7, it is clear that the predictive power of both fusion methods is high. In addition, we can qualitatively assess our decision fusion predictions as shown in Figure 8.

6. Conclusion and Future Work

Going forward, to improve model explainability and as an alternative fusion method, we are looking to implement coattention with the captions and images. We will also consider techniques for improving the performance of feature fusion, by incorporating emoji2vec embeddings to extract meaning from emojis, using a CNN architecture on the text documents, and doing further hyperparameter tuning.

Overall, the results of our multimodal classification frameworks are very promising, and beat state-of-the-art on this task.

7. Contributions

Both authors contributed equally to all parts of this work. All code can be found on GitHub.

References

[1] T. Amaratunga. Milestones of deep learning, 2017.

[2] S. B. Christos Dimitrakakis. Online policy adaptation for ensemble classifiers, 2004.

[3] Domo. Data never sleeps 5.0, 2018.

[4] M. A. Hussein Mouzannar, Yara Rizk. Damage identification in social media posts using multimodal deep learning, 2018.

[5] G. H. Laurens van der Maaten. Visualizing data using t-SNE, 2008.

[6] L. W. Pierre Geurts, Damien Ernst. Extremely randomized trees, 2006.

[7] J. Ramos. Using TF-IDF to determine word relevance in document queries, 2003.

[8] T. Shen. GRUs vs. LSTMs, 2017.

[9] L. Weng. Attention? attention!, 2018.

[10] T. Zahavy, A. Magnani, A. Krishnan, and S. Mannor. Is a picture worth a thousand words? A deep multi-modal fusion architecture for product classification in e-commerce. *CoRR*, abs/1611.09534, 2016.

[11] Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. Taylor Francis, 2012.