Multimodal Learning for Disaster Assessment Using Social Media Data

Swetha Revanur
Stanford University
srevanur@stanford.edu

Keanu Spies
Stanford University
keanus@stanford.edu

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 [1], 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 assess their confidence in crowdsourced media, given the susceptibility of the platforms to spammers and adversarial attacks. In social media analyses, computationally expensive multimodal approaches often fall to the wayside 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.

2. Infrastructure

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 [2]. Each image and its corresponding caption has one of six damage labels: fires, floods, natural landscape, infrastructural, human, and non-damage.

Specifically, our goal is to demonstrate that a multimodal approach yields higher damage classification performance compared to text-only or image-only baselines. We evaluate performance with a number of metrics including accuracy, precision, recall, and F1 score. In this milestone, we describe our data preprocessing steps and a text-only baseline.

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.

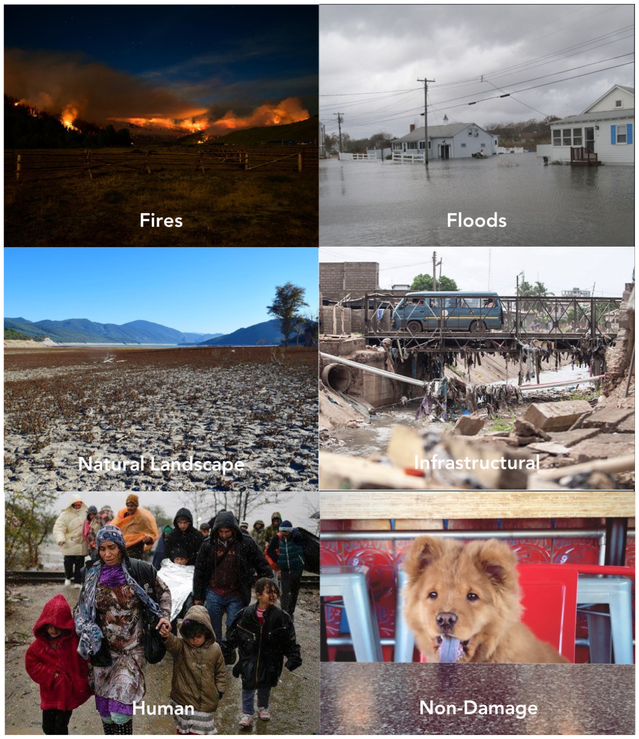

Figure 1. Selected images from each class.

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 Embedding (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 [3].

We can also use t-SNE without class labels, instead determining cluster membership with unsupervised k-means

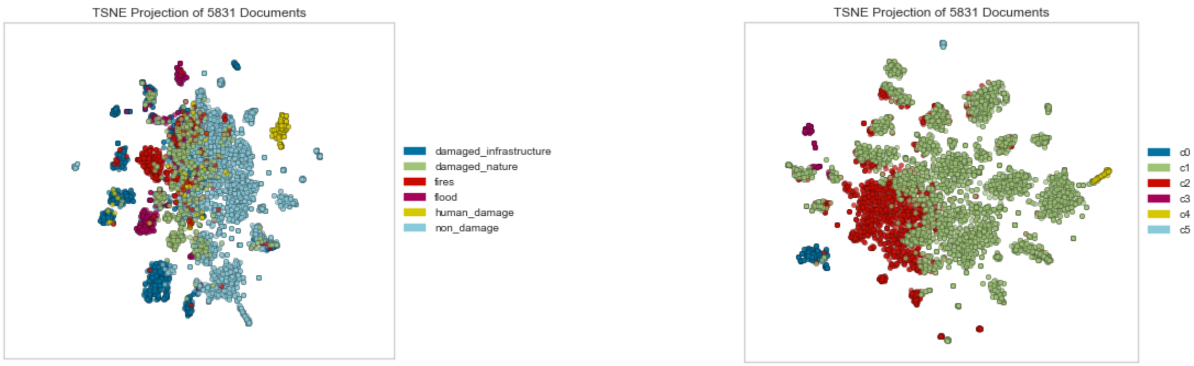

Figure 2. Left: t-SNE results with document labels; Right: t-SNE results after k-means clustering.

clustering. This will allow us to visualize clusters of related text based on content. Both t-SNE visualizations are shown in Figure 2.

3. Methods

3.1. Unimodal Baseline Approaches

To demonstrate the utility of multimodal learning, we establish unimodal baselines for comparison, starting with text-only approaches. In particular, we explore multinomial naive Bayes, Bernoulli naive Bayes, linear kernel SVM, 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 [4]. We use Word2Vec word embeddings as inputs into these classification 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 documentation. tf-idf reflects the importance of a given word in a document in a corpus [5]. Results are outlined in Table 1.

model	test_prec	test_rec	test_f1	test_acc
svm_tf	0.8942	0.8256	0.8585	0.9052
svm	0.8565	0.8208	0.8383	0.8906
multi_nb	0.8769	0.6940	0.7748	0.8472
multi_nb_tf	0.8882	0.4280	0.5777	0.7098
tree_tf	0.6037	0.4872	0.5392	0.6906
tree	0.5755	0.4687	0.5167	0.6853
bern_nb	0.4786	0.2845	0.3569	0.6680
bern_nb_tf	0.4786	0.2845	0.3569	0.6680

Table 1. Baseline metrics sorted by descending test accuracy.

We see that the linear kernel SVM with tf-idf-weighted word vectors performs the best, with a test accuracy just above 90%, and relatively high F1 scores, indicating class balance. 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. Interestingly, extra-trees with tf-idf weighting had the highest training time of 13.2820s, followed by SVM with tf-idf at 6.9583s. At inference time, SVM with tf-idf took 2.5922s, more than any other model.

4. Conclusions and Future Work

The unimodal models trained thus far show promise in successfully classifying captions. We will consider techniques for improving their performance, including deep neural networks, emoji2vec implementations to extract meaning from the emojis, text translation to English, in addition to image-only models and multimodal ones.

5. Contributions

Both authors contributed equally to this work.

References

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

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

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

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

[5] J. Ramos. Using tf-idf to determine word relevance in document queries, 2003.