Multi-Modal Training Data Creation with Snorkel [*]

Dmitry Kazhdan

January 7, 2019

Abstract

Existing Deep Learning (DL) approaches require large sets of labelled data to operate efficiently, which have to be produced by domain experts. This is a time-consuming, error-prone and expensive process, which is often regarded as the primary bottleneck in most DL applications. The recently introduced Snorkel system provides a way of dealing with this issue by relying on automated data label generation. However, all Snorkel-related work to date deals exclusively with textual input data. This report explores the possibilities of using Snorkel with other data modalities and labelling function types, thus further demonstrating its wide applicability.

1 Introduction

Machine Learning (ML) has achieved groundbreaking results in a wide variety of fields [11, 10, 6], and is currently one of the most active research areas in Artificial Intelligence. Many of these successes can be attributed to novel applications of Deep Learning (DL). DL techniques are capable of automatic feature extraction given a set of training data, thus relieving the developers from the burden of feature engineering. A subfield of ML that has extraordinary potential and is experiencing a resurgence of interest is *Multi-Modal Machine Learning* (MML). MML aims to build DL models that can process and relate information from multiple modalities (e.g. sound, image or text) by fusing them together, and has a wide range of possible applications [4, 18, 21] (a comprehensive overview is given in [3]). Unfortunately, DL techniques require large sets of labelled data to operate efficiently, which have to be produced by domain experts. This is a time-consuming, error-prone and expensive process, which is often regarded as the primary bottleneck in most DL applications.

Snorkel [23] is a first-of-its-kind system that was designed to address this problem by avoiding the necessity of manual data annotation. Snorkel makes use of the *data programming* paradigm [1, 19], where data annotation is automated by developing *labelling functions* (LFs) that express various heuristic knowledge. Whilst showing highly promising results, Snorkel was only applied to textual datasets in the original work.

The aim of this report is to show how Snorkel can be applied in a multi-modal context, thus further demonstrating its wide applicability. The contributions of this report are the following:

1. Showing how Snorkel may be used with a non-textual modality.
2. Showing how Snorkel may be used with a multi-modal dataset consisting of textually-annotated images.
3. Using unsupervised learning approaches for constructing LFs.

[*]Report word count: 2495

Section 2 gives an overview of the Snorkel system. Section 3 describes the multi-modal application used in this report. Implementation and evaluation are described in Section 4 and Section 5, respectively. Section 6 includes a discussion of possible directions for future work. Finally, Section 7 gives some concluding remarks.

2 Snorkel

Snorkel is a system capable of combining data labels from many weak supervision sources (cheaper sources of labels that have limited coverage and/or accuracy) to produce aggregated label values with improved accuracy and coverage. In doing so, Snorkel has to resolve conflicts amongst sources that assign different labels to the same data point, and determine which sources give higher accuracy.

Snorkel's workflow is comprised of three stages:

1. **Producing Labelling Functions:** The user is required to write labelling functions using Snorkel's interface. Snorkel represents data using a *context hierarchy* structure, which consists of a set of *context types* (a component of the data, such as an image, or a document), connected by parent-child relationships. A data point (referred to as a *candidate*) is defined as a tuple of contexts. A LF "λ" is defined as $\lambda : X \to Y \cup \{\emptyset\}$, where X is the set of input data points, Y is the set of output labels, and $\{\emptyset\}$ represents that the function abstains. These labelling functions can be arbitrarily complex in theory.

2. **Modelling of Source Accuracies and Inter-Functional Correlations:** Through the use of a generative model [19], Snorkel is able to learn the accuracies and correlations of the provided LFs. Crucially, this step is performed without access to ground truth data, relying only on inter-functional agreements and disagreements. Snorkel also uses an optimiser that employs various heuristics to decide on when and at what granularity to model these accuracies and correlations. This allows Snorkel to find the best performance/accuracy tradeoff. For more mathematically rigorous descriptions, analyses and proof sketches, see [20, 1, 19].

3. **Generating Probabilistic Labels:** Using the results of the previous step, the produced set of LF outputs is synthesised into a set of probabilistic labels, one per data point. The labelled data points constitute a labelled training set, which may be used as input for discriminative models.

Figure 1 provides an overview of the Snorkel system (this figure is taken directly from [20]).

3 Multi-Modal Setting

The implementation discussed in this report relies on the dataset and models given in [15]. This section presents an overview of these models and the associated dataset, as well as their application domain.

An emerging goal of the humanitarian computing community is building automated systems for detecting and flagging social media posts for disaster and crisis data. Work in [15] describes the construction of a multimodal model and corresponding dataset that can be used to partially achieve this goal. More specifically, a multimodal DL framework is used to identify damage-related information from social media posts and classify filtered posts into multiple damage and disaster categories. This is achieved by using a combination of multiple pre-trained unimodal convolutional neural networks that extract features from raw text and images independently, and then using a classifier for labelling the posts

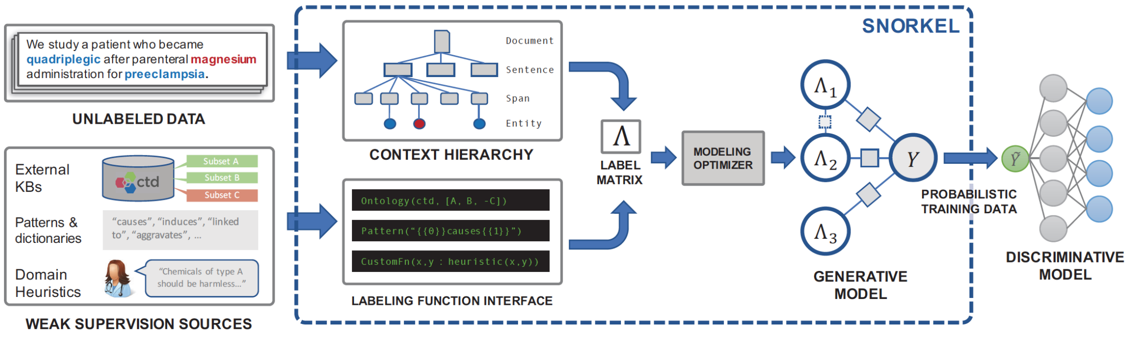
We study a patient who became quadriplegic after parenteral magnesium administration for preeclampsia.
UNLABELED DATA
External KBs
ctd
Subset A
Subset B
Subset C
Patterns & dictionaries
"causes", "induces", "linked to", "aggravates", ...
Domain Heuristics
"Chemicals of type A should be harmless..."
WEAK SUPERVISION SOURCES
SNORKEL
Document
Sentence
Span
Entity
CONTEXT HIERARCHY
Ontology(ctd, [A, B, -C])
Pattern("{{0}}causes{{1}}")
CustomFn(x,y : heuristic(x,y))
LABELING FUNCTION INTERFACE
Λ
LABEL MATRIX
MODELING OPTIMIZER
Λ1
Λ2
Λ3
Y
GENERATIVE MODEL
Ỹ
PROBABILISTIC TRAINING DATA
DISCRIMINATIVE MODEL

Figure 1: Overview of the Snorkel architecture

based on both modalities. This approach allows quickly identifying occurrences and types of various disasters and crises, which can then be used to contact first responders and aid them in allocating appropriate resources in response to such events (further details can be found in [15]).

Text classification is achieved by using a shallow CNN based on the architecture described in [14]. This CNN takes a matrix where each row is a real-valued vector representation of each word in the caption or tweet. The word-vectors are obtained from a pre-trained word embedding model. Image classification is achieved using the Inception CNN model [22], pre-trained on ImageNet. For both of these models, the last layer before the softmax layer is used to extract features that are then combined and used with a multimodal classifier.

An overview of the multi-modal model is given in Figure 2.

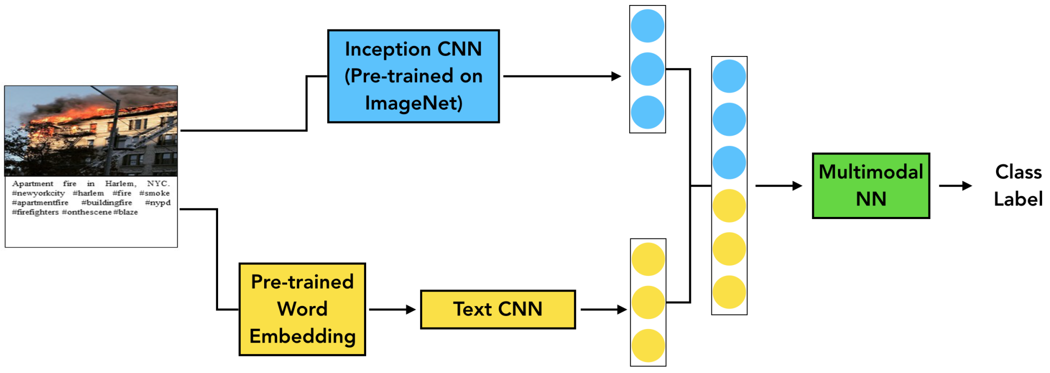
Apartment fire in Harlem, NYC. #newyorkcity #harlem #fire #smoke #apartmentfire #buildingfire #nypd #firefighters #onthescene #blaze
Inception CNN (Pre-trained on ImageNet)
Pre-trained Word Embedding
Text CNN
Multimodal NN
Class Label

Figure 2: Multimodal Architecture

The multi-modal dataset used to train and evaluate the model is openly-available from the UCI ML Repository [16], and consists of images with text, labelled with one of the following labels:

1. **Infrastructural**: damaged buildings, wrecked cars, and destroyed bridges
2. **Natural**: damage to natural landscape, such as landslides, avalanches, and falling trees

3. **Fires**: damage from wildfires and building fires
4. **Floods**: damage from city, urban and rural floods
5. **Human**: injuries and deaths
6. **Non Damage**: other images with text, besides the classes listed above

Source code used for setting up and training the relevant models is openly-available at [7]. For a more detailed discussion of the above system and related approaches, see [15].

4 Implementation

Snorkel was used to generate labels for the multi-modal dataset described in Section 3 (with the original hand-labels not seen during development). As in the original Snorkel paper [20], a small development set with hand-labels was used during LF construction. A total of 10 LFs was produced (1 text LF per class, and 4 clustering LFs). These LFs were then fed into Snorkel's generative model, as described in [20], in order to generate the training data labels.

4.1 Text Data

Text-processing LFs were written using techniques and approaches outlined in the following Snorkel tutorials: [5, 9, 8]. Since text was primarily in the form of very short descriptions and hash-tags, developing LFs that used sentence patterns, regular expression etc. was unnecessary. Instead, the developed LFs checked for presence of certain keywords. For every class, keywords were first selected based on common sense, and then refined using the development set. The refinement process revealed that it was more beneficial to employ a slightly different strategy for the non-damage class, where the function abstained if the text contained disaster-related words (the list of these words was acquired from section headings found in [17]). The produced set of text LFs is shown Figure 3.

```python
def mentions_fire(text_content):
    keywords = ['fire', 'flame', 'smoke', 'burn', 'burning', 'ash', 'forest', 'gas', 'explosion']

    for keyword in keywords:
        if keyword in text_content:
            return 1
    return 0

def mentions_damaged_flood(text_content):
    keywords = ['flood', 'water', 'rain', 'disaster', 'damage',]

    for keyword in keywords:
        if keyword in text_content:
            return 2
    return 0

def mentions_human_damage(text_content):
    keywords = ['death', 'dead', 'victim', 'loss', 'kill', 'injured',
                'injury', 'war', 'refugee', 'crime', 'attack', 'crisis', 'explosion']

    for keyword in keywords:
        if keyword in text_content:
            return 3
    return 0

def mentions_damaged_infrastructure(text_content):
    keywords = ['wreck', 'collapse', 'broken', 'bridge', 'building',
                'storm', 'disaster', 'damage', 'destroy', 'destruction']

    for keyword in keywords:
        if keyword in text_content:
            return 4
    return 0

def mentions_damaged_nature(text_content):
    keywords = ['nature', 'animals', 'wildlife', 'habitat', 'storm', 'destruction',
                'environment', 'tree', 'drought', 'landslide', 'forest', 'tornado']

    for keyword in keywords:
        if keyword in text_content:
            return 5
    return 0

def non_damage(text_content):

    bad_words = [
                'burn', 'wreck', 'collapse', 'disaster', 'damage', 'destroy', 'destruction',
                'death', 'dead', 'victim', 'kill', 'injured',
                'injury', 'war', 'refugee', 'attack', 'crisis',
                'storm', 'blizzard', 'drought', 'tornado', 'fire', 'landslide', 'earthquake', 'volcano',
                'eruption', 'flood', 'tsunami']

    for keyword in bad_words:
        if keyword in text_content:
            return 0
    return 6
```

Figure 3: Text-Processing Labelling Functions

Crucially, these LFs are noisy (since the selected keywords may be used in non-disaster contexts as well) and conflicting (keywords may appear in multiple LFs), developed using a very simple and intuitive methodology, as described above.

4.2 Image Data

Image processing LFs relied on an unsupervised clustering approach. Unfortunately, there were no semi-supervised clustering APIs capable of automatic cluster labelling available online. However, there were several unsupervised clustering APIs that could cluster a given set of images together, without attempting to meaningfully label the clusters themselves. One such open-source clustering API [12] was selected for this report. This API produced a hierarchical clustering of images and returned clusters at a given hierarchy level using a user-specified similarity threshold (more details can be found in [12]). The image clusters obtained using [12] were then labelled by hand. The corresponding LFs simply identified the cluster label from which the image came from (see [13] for the code). The above clustering approach was run with several different similarity thresholds (chosen values were 0.5, 0.6, 0.7 and 0.8), with one LF per clustering, thus utilising Snorkel's capability of managing potentially conflicting LFs. The above approach is summarized in Figure 4 (where LFs assign an integer identifier corresponding to a class label for every image).

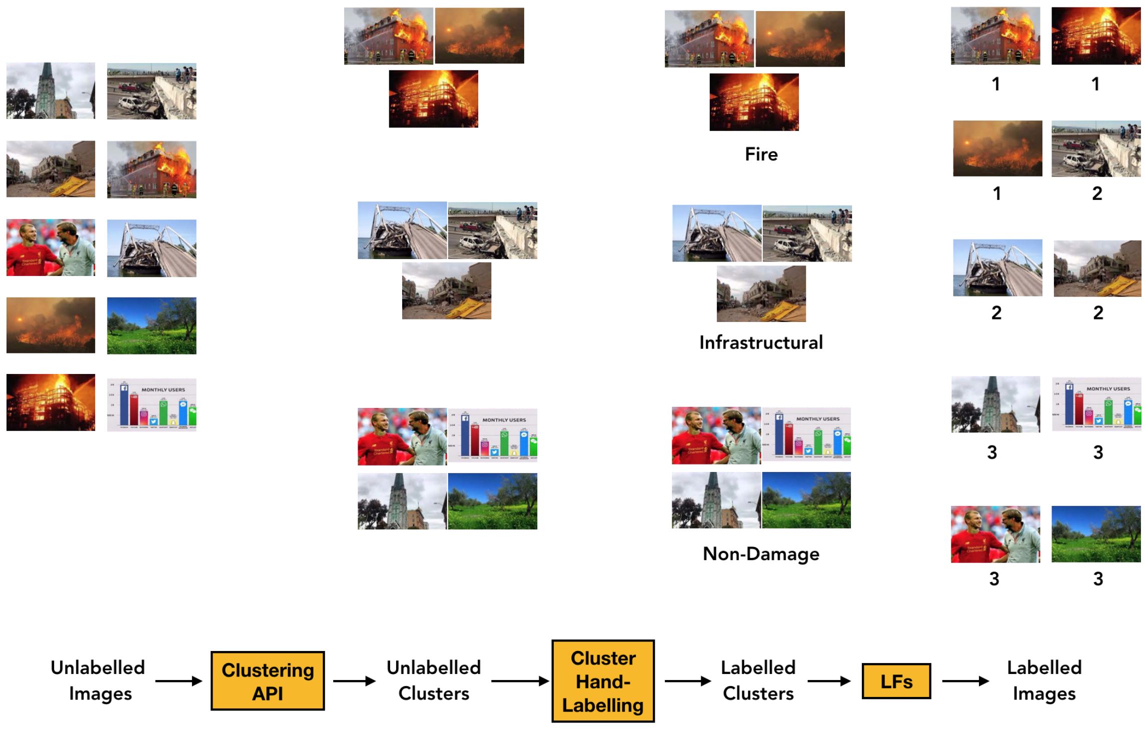

Figure 4: Image labelling example with 3 different classes

Crucially, the hand-labelling step was very quick to do, and differed from typical training data hand-labelling in a number of ways:

- The total number of clusters was $\sim$ 50, an order of magnitude smaller than the number of training samples.
- Cluster labelling did not have to be done as rigorously as typical data hand-labelling. Briefly observing the images contained in the cluster, and quickly deciding on which class the majority of these images likely belongs to was sufficient.

Overall, the above approach demonstrated how Snorkel may be used with image data, and how LFs may rely on unsupervised clustering approaches.

5 Evaluation

Snorkel-generated dataset labels were compared to the original hand-labels both directly, and by comparing performance of the multi-modal model presented in Section 3 after being trained on the multimodal dataset labelled with either label set. A comparison of how the unimodal and multimodal models were trained in this study as opposed to work in [15] is given in Appendix A.

5.1 Label Comparison

Firstly, the two sets of labels were compared directly, by computing the percentage of cases in which they assigned the same label to a data-point, as shown in Table 1. 'Image-only' and 'Text-only' rows correspond to using only image LFs or only text LFs (respectively) for label generation.

Labelling Functions	**Percentage Match**
Image-only	59%
Text-only	72.1%
Text and Image	79.6%

Table 1: Direct Label Comparison

Purely relying on the textual LFs produced a matching score of 72.1%. Image classification was noisier, attaining a lower matching score of 59%. Even at this stage, multi-modality achieved an improved result compared to unimodal approaches, increasing the matching score by 7.5%. Overall, this simple approach produced the same labels as the hand-labelling approach in nearly 80% of cases.

5.2 Model Performance Comparison

Secondly, the multi-modal model was trained and evaluated on a held-out test set of hand-labelled data, using the labelled datasets in turn. Table 2 shows a comparison of the aggregate model accuracy on the testing data obtained using the different dataset labels ('Original' refers to the original hand labels).

Labelling Functions	**Accuracy**
Text-only	76.1%
Text and Image	81.8%
Original	87.9 %

Table 2: Model Performance Comparison

As Table 2 shows, the dataset that used both text-based and image-based LFs improved on the text-only dataset by 5.7%, and was only 6.1% away from the hand-labelled one.

More detailed results were obtained by computing the **Precision**, **Recall** and **F1-Score** metrics for each class. Table 3 presents results for the hand-labelled dataset, and Table 4 presents results for the Snorkel-labelled dataset that used LFs of both modalities (together with differences from the hand-labelled one). The confusion matrices used to compute these metrics are given in Appendix A.

Precision was affected the most, dropping by 0.2, 0.14 and 0.12 for 3 of the 6 classes. **Recall** was more comparable, having a difference of 0.1 and below for 5 of the 6 classes. Overall, the **F1-Score** metric showed that even with the relatively simple, exploratory approach used in this work, Snorkel was still able to approach the performance of the hand-labelled approach, with score differences being at or below 0.08 for 4 of the 6 classes, and the largest difference being 0.13. This is further demonstrated in the confusion matrices, that are given in Appendix A.

Class	Precision	Recall	F1-Score
Infrastructure	0.69	0.78	0.73
Nature	0.93	0.65	0.76
Fire	0.91	0.85	0.88
Flood	0.81	0.87	0.84
Human	0.93	0.81	0.87
Non-Damage	0.92	0.98	0.95

Table 3: Hand-Labelled Dataset Metrics

Class	Precision	Recall	F1-Score
Infrastructure	0.73 (+0.04)	0.58 (-0.2)	0.65 (-0.08)
Nature	0.73 (-0.2)	0.55 (-0.1)	0.63 (-0.13)
Fire	0.90 (-0.01)	0.88 (+0.03)	0.89 (+0.01)
Flood	0.67 (-0.14)	0.78 (-0.09)	0.72 (-0.12)
Human	0.81 (-0.12)	0.79 (-0.02)	0.80 (-0.07)
Non-Damage	0.87 (-0.05)	0.94 (-0.04)	0.91 (-0.04)

Table 4: Snorkel-Labelled Dataset Metrics

6 Future Work

The primary focus of this study was to experiment with novel ways of applying Snorkel to multimodal datasets and comparing them to the hand-labelled approach, implying that relative performance was important, as opposed to absolute performance. However, re-running the above setup with larger training dataset sizes, longer training time, more rigorous parameter tuning (e.g. using cross-validation) etc. would most likely improve absolute performance of both approaches, making the benefits of these approaches even more significant.

Future work may also focus on improving the described clustering approach by exploring automated semi-supervised clustering methods, such as those described in [2]. The iterative nature of these algorithms typically makes them more accurate than the single-pass approach used here. Using a development set for semi-supervised labelling, and perhaps even using the other modalities for bootstrapping could eliminate the need for manual cluster labelling altogether, thus making the approach more scalable. Unfortunately, no suitable open-source implementations of semi-supervised image clustering were found, and thus a slightly more simplistic approach had to be adopted in this case.

Apart from relying on Snorkel's generative model, future work may also explore how to apply Snorkel's context hierarchy to other data modalities. In this case, it would require choosing a suitable hierarchical representation for images. Progress in this area could offer new ways of applying more types of LFs to other modalities by making efficient use of these context hierarchies.

Finally, there are numerous other applications besides disaster recovery that make use of multi-modal data processing and could benefit from the approaches discussed in this report. An important example is healthcare, where tasks often rely on fusing together data of multiple modalities (e.g. tomography scans and doctor's notes). Future work may explore how Snorkel may be applied in these domains, thus further demonstrating it's wide applicability.

7 Conclusion

This report successfully explored novel ways in which Snorkel may be used for multimodal dataset labelling. Snorkel was used to train several different state-of-the-art models relying on different data types that were combined to form a multi-modal model. Furthermore Snorkel was used with unsupervised clustering for image labelling. The approach presented in this report produced results that did not substantially differ from those obtained using the hand-labelling approach (despite relying on a number of simplifications), thus demonstrating Snorkel's suitability for multi-modal data label generation.

To conclude, Snorkel is a highly promising concept that simplifies many existing approaches to data generation and introduces new ones, changing the way users interact with ML systems. This report shows that Snorkel is applicable in a wide range of ML scenarios (e.g. unsupervised learning, multi-modal label generation etc.), and that there is still a great deal of untapped potential which can be explored in future work (as discussed in Section 6). Snorkel will likely be used with a wider range of ML applications in the future, reducing reliance on hand-labelling in preference to labelling using the *data-programming* paradigm.

Acknowledgements

This work made use of the following resources: [7, 16, 12, 23]. The author would like to thank the relevant contributors for making these resources available online and including appropriate supporting documentation.

References

[1] Stephen H. Bach et al. "Learning the Structure of Generative Models without Labeled Data". In: *CoRR* abs/1703.00854 (2017). arXiv: `1703.00854`. URL: `http://arxiv.org/abs/1703.00854`.

[2] Eric Bair. "Semi-supervised clustering methods". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 5.5 (2013), pp. 349–361.

[3] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal Machine Learning: A Survey and Taxonomy". In: *CoRR* abs/1705.09406 (2017). arXiv: `1705.09406`. URL: `http://arxiv.org/abs/1705.09406`.

[4] Catalina Cangea, Petar Velickovic, and Pietro Lio. "XFlow: 1D-2D Cross-modal Deep Neural Networks for Audiovisual Classification". In: *arXiv preprint arXiv:1709.00572* (2017).

[5] *Categorical Variables in Snorkel.* URL: `https://github.com/HazyResearch/snorkel/blob/master/tutorials/advanced/Categorical_Classes.ipynb`.

[6] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *CoRR* abs/1406.1078 (2014). arXiv: `1406.1078`. URL: `http://arxiv.org/abs/1406.1078`.

[7] *Damage Identification in Social Media Posts using Multimodal Deep Learning.* URL: `https://github.com/husseinmouzannar/multimodal-deep-learning-for-disaster-response`.

[8] *Extracting Spouse Relations from the News.* URL: `https://github.com/HazyResearch/snorkel/tree/master/tutorials/intro`.

[9] *Generating Weak Labels for Image Datasets (e.g. Person Riding Bike).* URL: `https://github.com/HazyResearch/snorkel/blob/master/tutorials/images/Images_Tutorial.ipynb`.

[10] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* (2015).

[12] *Image Clustering API.* URL: `https://github.com/elcorto/imagecluster`.

[13] Dmitry Kazhdan. *Multi Modal Training Data Creation with Snorkel.* URL: `https://github.com/dmitrykazhdan/Multi-Modal-Training-Data-Creation-with-Snorkel`.

[14] Yoon Kim. “Convolutional Neural Networks for Sentence Classification”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1746–1751. DOI: `10.3115/v1/D14-1181`. URL: `http://aclweb.org/anthology/D14-1181`.

[15] Hussein Mouzannar, Yara Rizk, and Mariette Awad. “Damage Identification in Social Media Posts using Multimodal Deep Learning”. In: *The 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM).* Rochester, USA, May 2018, pp. 529–543.

[16] *Multimodal Damage Identification for Humanitarian Computing Data Set.* URL: `https://archive.ics.uci.edu/ml/datasets/Multimodal+Damage+Identification+for+Humanitarian+Computing`.

[17] *Natural disaster.* URL: `https://en.wikipedia.org/wiki/Natural_disaster`.

[18] Jiquan Ngiam et al. “Multimodal Deep Learning”. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning.* ICML’11. Bellevue, Washington, USA: Omnipress, 2011, pp. 689–696. ISBN: 978-1-4503-0619-5. URL: `http://dl.acm.org/citation.cfm?id=3104482.3104569`.

[19] Alexander J Ratner et al. “Data programming: Creating large training sets, quickly”. In: *Advances in neural information processing systems.* 2016, pp. 3567–3575.

[20] Alexander Ratner et al. “Snorkel: Rapid Training Data Creation with Weak Supervision”. In: *CoRR* abs/1711.10160 (2017). arXiv: `1711.10160`. URL: `http://arxiv.org/abs/1711.10160`.

[21] Nitish Srivastava and Ruslan R Salakhutdinov. “Multimodal Learning with Deep Boltzmann Machines”. In: *Advances in Neural Information Processing Systems 25.* Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 2222–2230. URL: `http://papers.nips.cc/paper/4683-multimodal-learning-with-deep-boltzmann-machines.pdf`.

[22] Christian Szegedy et al. “Going deeper with convolutions”. In: June 2015, pp. 1–9. DOI: `10.1109/CVPR.2015.7298594`.

[23] HazyResearch Team. *Snorkel Website.* URL: `https://hazyresearch.github.io/snorkel/`.

A Appendices

A.1 Preprocessing and Training

The following pre-processing steps were performed on the original dataset [16]:

- To obtain a more balanced dataset, 300 samples were randomly selected from the **Infrastructural**, **Fire**, **Flood** and **Nature** classes, all of the 240 samples were selected from the **Human Damage** class, and 2000 samples were randomly selected from the **Non-Damage** class.
- The samples were randomly split into a non-test set (80% of samples) and test set (20% of samples). The non-test was then further randomly split into a training set (85% of samples) and validation set (15% of samples, referred to as the *development set*).

The image and text unimodal models were trained using the text and image training and validation data. The multimodal classifier (whose input is a concatenation of features from the penultimate layers of the unimodal networks) was trained on the same training set, using the pre-trained unimodal models.

The above pre-processing and training approaches differ from the approaches described in [16], meaning that accuracy results are not directly comparable. This is due to the fact that work in [16] made use of extra datasets (including separate unimodal datasets for submodel pre-training), which were not made openly-available.

A.2 Confusion Matrices

Table 5 and Table 6 present the confusion matrices (obtained by running the multimodal classifier on held-out test data) that were used to compute the relevant performance metrics given in Section 5.

Pred. / Act.	**Inf.**	**Nat.**	**Fir.**	**Flo.**	**Hum.**	**N.D.**	**Total**
Inf.	47	1	2	4	1	5	60
Nat.	8	39	2	6	2	3	60
Fir.	5	0	51	0	0	4	60
Flo.	3	2	0	52	0	3	60
Hum.	4	0	0	0	39	5	48
N.D.	1	0	1	2	0	236	240

Table 5: Hand-Labelled Training Data Confusion Matrix

Pred. / Act.	**Inf.**	**Nat.**	**Fir.**	**Flo.**	**Hum.**	**N.D.**	**Total**
Inf.	35	3	2	5	5	10	60
Nat.	8	33	1	9	2	7	60
Fir.	2	1	53	0	1	3	60
Flo.	1	3	0	47	0	9	60
Hum.	1	0	1	4	38	4	48
N.D.	1	5	2	5	1	226	240

Table 6: Snorkel-Labelled Training Data Confusion Matrix