Domain Adaptation for Disaster Tweets classification
Final Project Report

Roman Castagné
roman.castagne@gmail.com

Abstract

Disaster Tweet Classification is the task of categorizing tweets written during the time of a natural disaster or other important event, in order to extract information from insiders that could benefit emergency and rescue services. However, it is an inherently difficult task due to the amount of noise present in social media data. Possibly because of this noise, previous works have failed at transferring models trained on some datasets to new datasets containing different events, even when the events were related in nature. In this paper, we propose to evaluate this performance drop by designing a classifier for the task, then use a domain adaptation technique to improve the robustness of the model. Our work is available at https://github.com/RomanCast/DisasterTweetsClassification.

1. Introduction

Although Neural Networks have excellent generalisation abilities, they sometimes fail unexpectedly when applied to a different setting or domain than the original training domain. One example of such degradation is witnessed when training a Disaster Tweet Classification model. Disaster Tweet Classification aims at categorizing tweets written during a natural disaster (hurricane, earthquake etc.) between several categories, including non informative tweets (see figure 1 for an example). However, authors noted that models trained on a particular disaster struggled at inference on new, unseen disasters [Padhee et al., 2020]. This is a very concerning point in practice, since such a failure point could impact the work of emergency services that rely on it. In addition, each disaster has different characteristics and thus a different data distribution, with specific words related to each event for instance. In general, getting a domain shift between the training and the application of a machine learning model is usual.

The degradation in performance of the models under a domain shift is a common issue with neural networks. A large panel of methods, called Domain Adaptation, has been developed to solve these problems. In particular, such methods have been applied in the context of Natural Language Processing (NLP) [Ramponi and Plank, 2020].

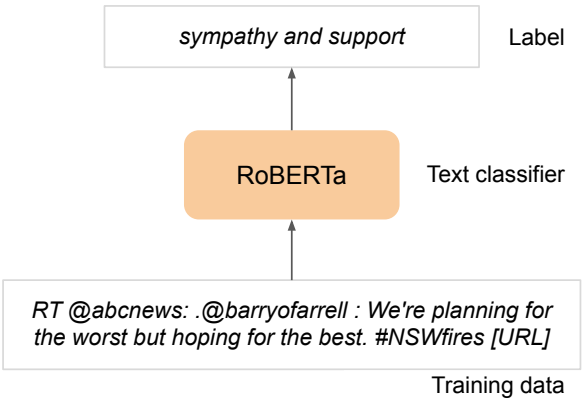

Figure 1. Overview of our simple text classifier used for Disaster Tweet Classification.

In this paper, we tackle the problem of limiting the performance drop of a disaster tweet classification model when applied on different data. We combine recent language models [Devlin et al., 2018, Liu et al., 2019] with a domain adaptation method called Domain Adversarial training of Neural Networks (DANN) [Ganin et al., 2016]. This method can be applied without important modifications to the model, allowing us to retain good classification performance.

Our contributions are the following:

- We train a classifier for disaster tweet classification in a general case, where we consider both important tweets and non-humanitarian tweets.
- We quantify the performance drop of our classifier when applying it to new domains, and try to explain this failure.
- We use a simple algorithm to make our model robust to data drawn from a different target distribution with unseen labels without any performance loss on the data from the source distribution.

We first define clearly our problem, then present an overview of methods for text classification in the context of disaster tweets. We then present our own work and the results obtained in our study.

2. Problem Definition

Given data points and labels $\{x_i^s, y_i^s\}_{i \in [\![1, n_s]\!]}$ sampled from a domain $\mathcal{D}_s$, we aim at fitting a model f minimizing an empirical risk $\mathcal{L}(f(x), y)$. In our case, the data is text from individual tweets and labels correspond to the category in which this tweet belong. We also consider that we have unlabeled data sampled from a different domain $\mathcal{D}_t$, $\{x_i^t, y_i^t\}_{i \in [\![1, n_t]\!]}$. In our case, this data could be tweets written during a different disaster, collected at a different moment or in a different region, tweets written with a local dialect, etc. We consider that we can only access the features x_i^t from this data and not the corresponding labels. Indeed, it is relatively easy to collect massive amounts of tweets, much harder to label them with correct classes.

We would like our classifier f trained on samples from $\mathcal{D}_s$ to perform similarly on samples from the target domain $\mathcal{D}_t$. Techniques from the field of unsupervised domain adaptation aim at reaching such goal by using unsupervised information $\{x_i^t\}_{i \in [\![1, n_t]\!]}$ during the training of the classifier.

3. Related Work

3.1. Disaster Tweet Classification

The task of Disaster Tweet Classification requires to categorize tweets written at the time of an event (natural disaster, terrorism, etc.). These tweets may then be used to inform rescue teams on the state of infrastructures, medical teams about injured persons, or help raise funding to cope with damages. This classification also requires to filter out non-humanitarian tweets, written at the same time as the event. This data is hard to process since it constitutes the major part of the information flow and is not related to a specific subject. Some studies choose to remove those tweets when preprocessing the data [Padhee et al., 2020] to concentrate on the actual classification of disaster-related tweets. Other studies consider a binary classification setting, where the goal is to distinguish non-humanitarian tweets from the rest [Li and Caragea, 2020].

One of the major issues with this classification task is to deal with the abundance of domains, with each disaster generating tweets from a different text distribution. Datasets published over the years have aggregated information from several events each time [Imran et al., 2016, Alam et al., 2018] but differ in the quality, the language of the data, and the annotations collected. Recently, [Alam et al., 2020] have assembled several datasets for disaster tweet classification, cleaned them and tagged languages for each tweet, making it an easy-to-use global resource.

3.2. Domain Adaptation

Domain Adaptation aims at improving the robustness of models when presented out-of-domain data, generated from a different distribution. This problem is important in Machine Learning, where we are unable to evaluate how a model would perform outside of its training and testing set. Neural networks have excellent generalisation abilities for in-domain data, but sometimes lack robustness when dealing with new sources of data. For instance, Transfer Learning has helped improve the general performance of models in Vision and NLP [Devlin et al., 2018] with greatly reduced costs of fine-tuning by pretraining on general domains, e.g. Wikipedia. However, adapting the same models to new domains (e.g. Medical data) is a difficult task [Lee et al., 2020].

[Ramponi and Plank, 2020] dress an overview of existing domain adaptation techniques for NLP, and distinguish two classes of methods: data centric domain adaptation, where the algorithms aim at modifying the distribution of the data itself to avoid skewing the model, and model centric domain adaptation, where we modify the model itself to improve its robustness. The technique we chose to use in this paper belongs to the last category.

Domain Adversarial training for Neural Networks [Ganin et al., 2016] introduces a new loss that optimises for robustness, by using the original label predictor and an additional "domain classifier". Formally, it decomposes the classifier f in $G_y \circ G_f$ with G_f a feature extractor and G_y the label predictor, which can be a simple two layer perceptron mapping features to labels. The classification loss can be written as:

$$\mathcal{L}_y = \frac{1}{n_s} \sum_{i=1}^{n_s} l(G_y \circ G_f(x_i^s), y_i^s) \tag{1}$$

Another small classifier tries to distinguish between source domain data (label s), and target domain data (label t) and is optimized to fail at this task, in order to leverage model representations that are domain agnostic. The idea is to align model representations from both domains. That second classifier $G_d \circ G_f$ is optimised maximising the loss:

$$\begin{aligned} \mathcal{L}_d = &\frac{1}{n_s} \sum_{i=1}^{n_s} l(G_d \circ G_f(x_i^s), s) \\ &+ \frac{1}{n_t} \sum_{i=1}^{n_t} l(G_d \circ G_f(x_i^t), t) \end{aligned} \tag{2}$$

In the case of Transfer Learning for NLP, the feature extractor G_f can be a pretrained model, e.g.

BERT[Devlin et al., 2018]. The final loss to optimise is the following, with α a parameter regularising the domain classifier loss:

$$\mathcal{L} = \mathcal{L}_y - \alpha\mathcal{L}_d \tag{3}$$

At training time, we can use the gradients of the model like in standard backpropagation to optimise our network. It requires to add a gradient reversal layer that will inverse the signs of the gradients of the domain classifier and multiply these by the constant α.

An important advantage of DANN resides in the fact that it only requires minimal changes to the model. With almost no additional parameters (except for the domain classifier G_d) the model is not heavier. Furthermore, no assumptions are made on the feature extractor, which can then be chosen to be any powerful model from the literature.

The main caveat resides in the fact that it requires unlabeled data sampled from the target domain. This data is accessible in our case, but new data might not lie within our original target distribution, and the kind of robustness the model has acquired for this distribution might be useless in front of that novel distribution. Still, we argue that training on enough unlabeled data should help.

4. Methodology

The first step in improving the robustness of the model was to understand how the different datasets and events relate to each other, how the labels are distributed across examples, what kind of data is studied, etc. We present this analysis in a first part, then describe the vanilla model we used as well as the model trained to optimise for robustness. Finally, to ensure reproducibility of our results, we expose our hyperparameters and other training details in a third part.

4.1. Dataset

We used the dataset of [Alam et al., 2020] to conduct our study, available freely online[1]. This dataset is a concatenated and cleaned version of eight disaster tweet classification datasets, including CrisisNLP [Imran et al., 2016], CrisisMMD [Alam et al., 2018] and CrisisLex [Olteanu et al., 2014]. It spans over 60 different events, contains 11 class labels and 70,192 data points. The number of samples per source varies greatly, the largest being CrisisLex (29,150 examples) and CrisisNLP (21,866).

We present in table 1 some examples of tweets written with their corresponding labels from the `hurricane_irma` event in the CrisisMMD dataset.

[1] https://crisisnlp.qcri.org/crisis_datasets_benchmarks.html

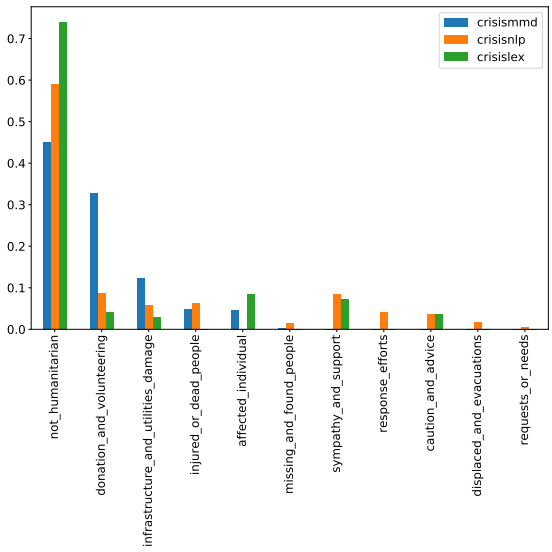

Figure 2. Proportion of examples per class for each of the major datasets

We can see that this data is quite noisy, often written in an unconventional English.

We plot the distribution of the data across labels for three of the major datasets in figure 2. These distributions are very imbalanced: due to the abundance of tweets being published at every moment, `not_humanitarian` labels constitute the majority of the data. For instance, they represent more than 70% of examples in CrisisLex, and 60% of examples in CrisisNLP.

The second observation is that labels do not exactly match across datasets. For instance, CrisisMMD does not have any label for `sympathy_and_support`, CrisisLex has no label for `injured_or_dead_people`, and only CrisisNLP has examples for the label `response_efforts`. We computed the maximum accuracy obtainable by a system trained on one of these three datasets and evaluated on another dataset. We found that this mismatch actually did not impact the maximum obtainable accuracy too much. The worst case is a model trained on CrisisNLP and evaluated on CrisisMMD, that could only reach 81.48% accuracy. In practice, the actual accuracy is often way lower than the maximum obtainable accuracy.

4.2. Model

Following [Padhee et al., 2020], our initial model is RoBERTa [Liu et al., 2019], a pretrained Transformer model. We used the `roberta-base` version from the HuggingFace library [Wolf et al., 2019]. This version possesses 125 million parameters, with 12 layers each containing 12 heads, and hidden size of 768. We use the default

Tweet	Label
RT @TheTrackAddictz: Come hangout with me ! Hurricane Irma on #BIGOLIVE [URL]	not_humanitarian
Residents/staff are beginning return to @goodsam - Kissimmee Village in the wake of flooding after #HurricaneIrma.... [URL]	affected_individual
Warning: many school zone signs are down or inoperative due to Irma. [URL] [URL]	infrastructures_and_utilities_damage
RT @bertandpatty: Riding out Hurricane #Irma in Roseau #Dominica #travel [URL]	donation_and_volunteering

Table 1. Examples of tweets written during the event `hurricane_irma`, and the corresponding labels. We replaced URLs by a tag [URL] for the table.

dropout probability when training, 0.1.

The model trained with DANN uses the same encoder than the vanilla model, RoBERTa. We use the same classification layer as well, a two layer perceptron with the same hidden size of 768.

4.3. Training Details

We preprocessed the examples following [Alam et al., 2020] by removing mentions ("RT", "@..."), keeping only ASCII characters, removing URLs, removing the "#" character, and replacing consecutive spaces with a single one. We then split the data into training and test sets using 80% of the original data for training.

We trained the models for a maximum of 10 epochs on a single Tesla T4 GPU card with 16Gb of memory, accessible through Google Colab. We used a batch size of 32 for our experiments. We used AdamW [Loshchilov and Hutter, 2018] with a learning rate of 1×10^{-5}. Training took at most one hour with this configuration.

5. Results

In this section, we first expose the results we obtained with our text classifier, the performance of the models, and the drop in F1 scores when evaluating those classifiers on different events or data sources. We then show that training the classifier adversarially does not hurt its original performance, and compare its results evaluated on out-of-domain data against the vanilla text classification model.

5.1. Performance drop in text classification

Contrary to the systems from the literature, we train a model without distinguishing the `not_humanitarian` label from the rest. Other papers either transform the task into a binary classification (related or non-related tweet) or completely remove this label. In our case, we consider 11 different labels (including non-related tweets). For this reason, we have no baseline to compare to in our results. However, the actual performance of the classifier is not the aim of this study, but rather its robustness to new data.

Training event	Size	Accuracy	F1
SriLanka Floods	592	92.4	53.97
Hurricane Maria	2,191	78.6	61.74
Hurricane Harvey	2,050	84.4	76.67
Mexico Earthquake	656	84.9	69.81
California Wildfires	662	79.7	78.78

Table 2. Results of several vanilla RoBERTa model evaluated on the same events they were trained on.

Training source	Size	Accuracy	F1
CrisisMMD	8,063	79.7	65.81
CrisisNLP	21,866	80.7	61.91
CrisisLex	29,150	93.2	80.46

Table 3. Results of several vanilla RoBERTa model evaluated on the same sources they were trained on.

We present in table 2 and 3 our results on three different sources of data, as well as for several different events in the CrisisMMD dataset.

To quantify the drop in performance of these models when evaluated on other domains, we use each model trained on a single event or on a single data source (which encompasses several events) and evaluate it on another event or source. We plot the differences in F1 scores compared to the performance on in-domain data on figures 3 and 4. If we denote by $F1(.)$ the F1 score of a model and $D_0 \rightarrow D_1$ a model trained on domain D_0 and evaluated on domain D_1, the scores plotted are:

$$F1(D_0 \rightarrow D_1) - F1(D_1 \rightarrow D_1)$$

that is, the score the out-of-domain model obtains minus the score the in-domain model obtains.

For the *events*, we notice that we do not necessarily ob-

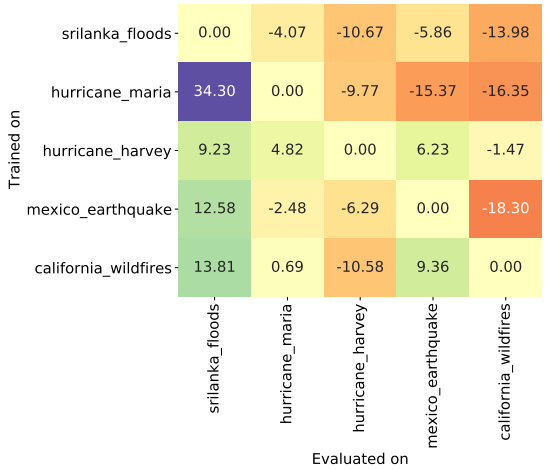

Figure 3. Performance of models trained on the event on the left and evaluated on the event below. The difference in F1 score between the in-domain evaluation and out-of-domain evaluation is plotted.

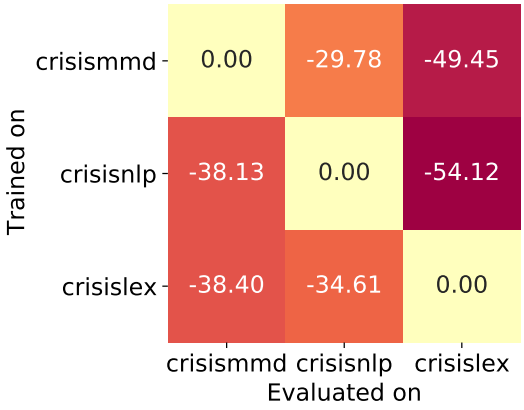

Figure 4. Performance of models trained on the data source on the left and evaluated on the data source below. The difference in F1 score between the in-domain evaluation and out-of-domain evaluation is plotted.

serve a drop in performances. For instance, almost all models improve on the original model when evaluated on Sri-Lanka floods. We hypothesize that this is probably due to the small number of examples in the SriLanka floods' event. Similarly, the model trained on the event Hurricane Harvey improves in most cases when evaluated on new events, which may be due to the high number of samples compared to other datasets. However, although the event Hurricane Harvey contains an important number of examples, it fails when evaluated on other datasets, even when the datasets are closely related like Hurricane Harvey.

However, the F1 scores degrade by up to 54 F1 points for the different *data sources*, indicating an complete lack of robustness to different data collection methods. The smallest degradation is still 29.78 F1 points. This could be due to different factors: different distribution of labels, different data cleaning methods done by the authors of these datasets, or very different events across the datasets.

5.2. Comparison with adversarial training

We now compare the F1 scores obtained by the adversarially trained classifier against the vanilla RoBERTa model in table 4. We evaluated the adversarial model exclusively in the case of different data sources, since the drops in performances in figure 4 are way higher than the drops in performances when using different events.

We can see from these results that although the adversarial training does not incur any loss in performance compared to the original model when evaluating on the same domain as the training data, it also fails at being more robust to new domains. The results when evaluating on a different domain are always within 2 F1 points from the results of the original model.

Source	Target	RoBERTa	AdvRoBERTa
CrisisMMD	CrisisMMD	65.81	**71.99**
CrisisMMD	CrisisNLP	27.68	27.84
CrisisMMD	CrisisLex	27.41	**27.77**
CrisisNLP	CrisisMMD	**32.13**	30.76
CrisisNLP	CrisisNLP	**61.91**	60.58
CrisisNLP	CrisisLex	27.3	**28.1**
CrisisLex	CrisisMMD	**31.01**	30.61
CrisisLex	CrisisNLP	26.34	26.28
CrisisLex	CrisisLex	80.46	**80.77**

Table 4. F1 scores of models evaluated on "Target" and trained on "Source". We denote the adversarially trained RoBERTa model by AdvRoBERTa. When the target domain is the same as the source domain, we take the best score obtained by AdvRoBERTa (of the two adversarial trainings).

Several factors might explain this failure to improve robustness. The first one is related to the tuning of the adversarial training. Further analysis would be needed to explore hyperparameters of the model, such as the number of training steps for this method, or searching for an optimal value of α, the parameter weighting the domain loss in the objective value of DANN. The second might be related to a misconception regarding the reason for the lack of robustness of the model. Indeed, when using DANN, we assume that the distribution of the features $x_{1:n}$ is shifting, while the shift might only concern the distribution of labels $y_{1:n}$. Adapting a model in this setting would be a different problem however, related to the application pipeline.

6. Conclusion

In this study, we propose to use a recent language model, RoBERTa, to build a text classifier categorizing tweets written during natural disasters in the hope of providing emergency services tools to react quickly and efficiently during such events. This task has been addressed in the literature but often is restricted to easier settings, such as a reduced number of labels. More importantly, it has been noted that such classifier is not robust to new domains, for instance when evaluated on different disasters, or on different data sources. We quantify this drop in performance with the help of our classifier, and observe that this drop is particularly important for different data sources, with losses up to 55 F1 points compared to the original model. Such unstable behaviour might prevent the use of these models in production.

In a second part, we proposed a model robust to out-of-domain data by training conjointly on unlabeled data from the out-of-domain distribution. This model combines the same encoder than in the first part, RoBERTa, but adds a domain classifier to encode representations agnostic to the domain from which the data is sampled. We evaluated this new classifier trained adversarially on different sources, and showed that this method failed at improving the robustness of the model, bringing only marginal gains over the original model when evaluating on out-of-domain data.

In future work, we propose to dive deeper into the failure of the adversarially trained model, in order to evaluate whether this failure is due to the domain adaptation technique being unfit, or another reason inherent to the data for example. Depending on these findings, we might spend some time tuning the adversarial model, or explore new domain adaptation techniques.

Another important domain shift is the language in which tweets are written. Future work should consider multilingual models and evaluate the impact of the language on the robustness of the model. Finally, an improvement to the task of Disaster Tweet Classification could be the integration of image data, present in a majority of tweets under the form of URLs, using multi-modal (i.e. text and language) models.

References

[Alam et al., 2018] Alam, F., Ofli, F., and Imran, M. (2018). Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*. 2, 3

[Alam et al., 2020] Alam, F., Sajjad, H., Imran, M., and Ofli, F. (2020). Standardizing and benchmarking crisis-related social media datasets for humanitarian information processing. *arXiv preprint arXiv:2004.06774*. 2, 3, 4

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 1, 2, 3

[Ganin et al., 2016] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030. 1, 2

[Imran et al., 2016] Imran, M., Mitra, P., and Castillo, C. (2016). Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA). 2, 3

[Lee et al., 2020] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240. 2

[Li and Caragea, 2020] Li, X. and Caragea, D. (2020). Domain adaptation with reconstruction for disaster tweet classification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1561–1564. 2

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*. 1, 3

[Loshchilov and Hutter, 2018] Loshchilov, I. and Hutter, F. (2018). Fixing weight decay regularization in adam. 4

[Olteanu et al., 2014] Olteanu, A., Castillo, C., Diaz, F., and Vieweg, S. (2014). Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8. 3

[Padhee et al., 2020] Padhee, S., Saha, T. K., Tetreault, J., and Jaimes, A. (2020). Clustering of social media messages for humanitarian aid response during crisis. *AI for Social Good, Harvard CRCS Workshop*. 1, 2, 3

[Ramponi and Plank, 2020] Ramponi, A. and Plank, B. (2020). Neural unsupervised domain adaptation in nlp—a survey. *Coling 2020*. 1, 2

[Wolf et al., 2019] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910. 3