Tweedr: Twitter for Disaster Response

Zahra Ashktorab Chris Brown Jit Nandi Aron Culotta

August 23, 2013

1 Introduction

I. Context

II. Problem

III. Solution Overview

2 Data

I. Unlabeled data from different disasters

II. Labeling for classification (and uniform vs keyword sampling)

III. Labeling for extraction

IV. Summary statistics (number labeled/unlabeled; number of each class; number by disaster)

We identified 12 crisis events that occurred in North America since the founding of Twitter in 2006. We then constructed queries to collect relevant tweets from Gnip, a social media aggregation company. We constructed two types of queries: (1) keyword (**kw**) queries contain search terms and hashtags determined to be relevant based on a post-hoc analysis; (2) geographical queries (**geo**) consisting of a bounding box of coordinates around the epicenter of the event. Table 1 lists the number of tweets collected for each event.

In a real-world deployment, it may be difficult to manually construct a list of keywords relevant to a disaster, but we leave this for future work.

2.1 Data Annotation

To train and evaluate our automated methods, we must first collect human-annotated examples. We consider two tasks for annotation:

1. **Classification**: Does the tweet mention either specific infrastructure damage or human casualty? We treat this as a binary classification task. Positive examples include "10 injured in plant explosion" and "The windows are smashed at the Whole Foods on 1st"; however, "Hurricane Irene causes massive damage" would be a negative example, since it does not include specific, actionable damage information.

2. **Extraction**: For positive examples of the above, identify the tokens in the tweet corresponding to specific types of infrastructure damage or counts of the number of dead or injured. For example, in the tweet "Flooding bad up and down sides of Green River Rd," the token "Flooding" should be annotated as a damage type, and the tokens "Green River Rd" should be labeled as a road. The full ontology is listed in Table [**FIXME**: add].

Since not all data can be labeled manually, we sample a small subset. Half of the tweets are selected uniformly at random from each event; the remaining half are sampled from tweets matching a set of keywords heuristically determined to be relevant to our task.[1] We do this to mitigate the class imbalance problem (i.e., most tweets are not relevant to infrastructure damage or casualties).

We sampled 1,049 tweets of the resulting tweets, of which 793 were labeled as positive examples. We then annotate the extraction labels for each positive example.

2.2 Classification

We compare a number of standard classification algorithms, including K-nearest neighbors, decision trees, naïve Bayes, and logistic regression, as implemented in the `scikit-learn` Python library.[2] We also compare with supervised latent Dirichlet allocation [1],

[1] The keywords are: bridge, intersection, car, bus, truck, vehicle, evacuation, evacuate, fire, police, institution, wind, impact, injured, damage, road, airplane, hospital, school, home, building, flood, collapse, death, casualty, missing.

[2] `http://http://scikit-learn.org/`

Event	N (kw)	N (geo)	total	keywords
christchurch	759,399	2,017	761,416	#EQNZ,#CHCH,#NZquake,Christchurch
ike	113,773	3,989	117,762	Hurricane+Ike,Hurricane,Ike,Galveston,Houston
irene	384,689	3,871,631	4,256,320	#Hurricane,#Irene,#Tropics
moore	842,318	1,579,056	2,421,374	#mooretornado,#moore,newcastle
oklahoma	1,697,891	6,209,510	7,907,401	#oklahoma,#tornado,#oklahomatornado,#okwx,#okc
samoa	235,208	2,016	237,224	Samoa,tsunami,earthquake
slavelake	45,072	1,281	46.353	#SlaveLake,Slave+Lake
supertuesday	22,205	2,201	24,406	Super+Tuesday,Jackson,Memphis,supertuesday
tornado2011a	476,730	2,293	479,023	Tushka,oklahoma,okwx,arkansas,akwx,tornado
tornado2011b	52,201	2,326	54,527	#alwx,#okwx,#txwx,#tristatewx,tornado,#ALNeeds, #ALHaves,#WeAreAlabama
vtech	16,652	2,869	19,521	#vatech,#virginiatech,#hokies,#vtech,#vt
westtx	651,045	178,846	829,891	#WestExplosion,#WestTX
Total	**5,297,183**	**11,858,035**	**17,155,218**	

Table 1: Number of tweets collected by event. We query for tweets both by keyword (**kw**) and geographical bounding box (**geo**).[**FIXME**: Sort by year or size?]

for which we create a Python wrapper of the R `lda` package.[3]

2.3 Clustering

We consider two different methods of approximate string matching: Bloom filters [2] and SimHash [3].

2.4 Extraction

For extraction, we use a conditional random field (CRF) [4], as implemented by the `CRFsuite` toolkit.[4]

3 Experiments

I. Classification results

   A. overall precision, recall, f1

   B. compared with predicting on unseen disasters

   C. comparison of sLDA and vanilla classifiers

   D. visualize important features (e.g., sLDA graph)

   E. list some exemplary good/bad classifications

II. Clustering results (maybe don't need accuracy, but at least what percent is duplicate)

III. Extraction

Correctly identified as damage or casualty
xxxx
Incorrectly identified as damage or casualty
xxxx

   A. overall precision, recall, f1, confusion matrix

   B. compared with predicting on unseen disasters

   C. visualize important features

   D. list some exemplary good/bad classifications

4 Related Work

- Extracting Information Nuggets from Disaster-Related Messages in Social Media
- Practical Extraction of Disaster-Relevant Information from Social Media
- Social Media Data Mining: A Social Network Analysis Of Tweets During The 2010-2011 Australian Floods
- TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief
- Natural Language Processing to the Rescue?: Extracting Situational Awareness Tweets During Mass Emergency

[3] `http://cran.r-project.org/web/packages/lda/`
[4] `http://www.chokkan.org/software/crfsuite/`

	All			New Disaster		
Method	**F1**	**Pr**	**Re**	**F1**	**Pre**	**Re**
sLDA	0.01 ± 0.10	0.01 ± 0.10	0.01 ± 0.10	0.01 ± 0.10	0.01 ± 0.10	0.01 ± 0.10
SVM	F1	Pr	Re	F1	Pre	Re
LogReg	F1	Pr	Re	F1	Pre	Re

Table 2: Classification results

	All			New Disaster		
Features	**F1**	**Pr**	**Re**	**F1**	**Pre**	**Re**
All	0.01 ± 0.10	0.01 ± 0.10	0.01 ± 0.10	0.01 ± 0.10	0.01 ± 0.10	0.01 ± 0.10
feature1	F1	Pr	Re	F1	Pre	Re
feature2	F1	Pr	Re	F1	Pre	Re

Table 3: Extraction results

5 Conclusions and Future Work

I. Summarize what we did

II. Mention limitations

III. Summarize next steps

6 Acknowledgements

This work was performed during the 2013 Eric & Wendy Schmidt Data Science for Social Good Fellowship at the University of Chicago, in partnership with the Qatar Computational Research Institute. We are greatful to Gnip for providing access to the historical tweets used in this analysis, as well as to all the 2013 DSSG Fellows who helped with data annotation.

References

[1] D. M. Blei and J. D. McAuliffe. Supervised topic models. arXiv e-print 1003.0783, Mar. 2010.

[2] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422426, July 1970.

[3] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, STOC '02, page 380388, New York, NY, USA, 2002. ACM.

[4] C. Sutton and A. K. McCallum. *An introduction to conditional random fields*. Now Publishers, Hanover, MA, 2012.