

Detection and Analysis of Disaster-Related Tweets

Daniel Solomon

solomond@mail.tau.ac.il

Gal Ron

galr1@mail.tau.ac.il

Omri Ben-Horin

omribenhorin@mail.tau.ac.il

Abstract

Twitter has become an important channel of communication during times of emergency and disasters; it allows individuals to publish user-created content written in colloquial language, and update others on events from around the world. In this paper we present our work on disaster-related tweets using tools from the NLP pipeline, such as part-of-speech (POS) tagging and named-entity recognition (NER). We used a publicly available labeled dataset of tweets, and integrated our own classification tools together with Twitter-specific publicly available NLP tools, to create a classifier that reaches 94% accuracy in identifying disaster-related tweets. We also present an additional classifier trained to distinguish between "objective" disaster-related tweets and "subjective" ones. Lastly, we used NER to extract information on disasters, and experimented with tweets recently sent from locations where natural disasters have occurred.

Our code is available at: <https://github.com/glrn/nlp-disaster-analysis>.

1 Introduction

The popular microblogging service Twitter is a fruitful source of user-created content. With hundreds of millions of new tweets every day, Twitter has become a probe to human behavior and opinions from around the globe. The Twitter 'corpus' reflects political and social trends, popular culture, global and local happenings, and more. In addition, tweets are easy to access and aggregate in real-time. Therefore, we experience an increased interest in natural language processing research of Twitter data.

As one of the world's most widely used social networks, Twitter is an effective channel of communication

and plays an important role during a crisis or emergency. The live stream of tweets can be used to identify reports and calls for help in emergency situations, such as accidents, violent crimes, natural disasters and terror attacks (which we all refer to as 'disasters' in this paper).

In this work we utilize techniques from the natural language processing pipeline (tokenization, part-of-speech tagging and named-entity recognition) to work on Twitter data, as opposed to traditional corpora, in order to detect and analyze disaster-related tweets.

The Dataset We present our experiments on a dataset of 10,877 tweets¹, labeled to '*disaster-related*' and '*not disaster-related*' with confidence in the range $[0, 1]$. For example, the following tweet is '*disaster-related*' with confidence 1,

Thunderstorms with little rain expected
in Central California. High fire danger.
#weather #cawx <http://t.co/A5GNzbuSq>

while the following tweet is '*not disaster-related*' with confidence 0.59,

It's been raining since you left me // Now I'm
drowning in the flood // You see I've always
been a fighter // But without you I give up

Even for one who is not familiar with the Bon Jovi lyrics in the latter tweet, it is clear that the tweet does not refer to a real natural disaster. However, this observation is hard to make examining only the vocabulary used; the tweet contains a variety of 'disastrous' words (raining, drowning, flood, fighter). This example hints that in order to reach meaningful results we may have to examine additional linguistic features of colloquial writing, as well as Twitter-specific features such as hashtags (#), user at-mentions (@), internet links and emoticons.

Our Contribution In this paper we present our work tackling three missions involving disaster-related tweets.

The first mission is *identification* of disaster-related tweets among a variety of tweets. We implemented several classifiers, the best of which achieved around 94% accuracy on the dataset. We note that this method could have easily been adjusted to identify tweets related to themes other than disasters (e.g. politics-related, sports-related, etc.), given the appropriate dataset. This part of our work is described in section 2.

The second mission is binary classification of disaster-related tweets to one of two categories, *subjective tweets* (i.e. tweets that express an emotion) vs. *objective tweets* (such as news reports on disasters). To achieve this we manually tagged 2,410 disaster-related tweets. The motivation behind this task is that objective tweets, such as informative news reports, are likely to be published after the event had already become clear to emergency services, while subjective tweets may contain invaluable first-person testimonies of ongoing events. This part of our work is described in section 3

Finally, we extracted named entities to enrich our knowledge on the disaster, comparing two NER methods (Twitter-oriented vs. general one). This part of our work is described in section 4.

To demonstrate our framework we aggregated recent tweets from various locations in the US, extracted disaster-related tweets using our classifier, and then used named-entity recognition to discover entities related to ongoing disasters. For example, "Hurricane Harvey" appeared as a top named-entity among recent tweets sent from Houston, TX, which we identified as *disaster-related*. Further examples of results on recent data are presented in section 5.

1.1 Twitter vs. Traditional Corpora

Tweets are limited to 140 characters and are widely used by non-professional writers. Therefore, Tweet datasets have some unique features that differ from traditional corpora (such as WSJ corpus). These features should be addressed when implementing natural language processing techniques.

First, the grammar of tweets is quite different from edited news text. It is common that tweets are written as colloquial sentences in first person where the subject ('I') is omitted, as in: 'see the flames from my window OMG'.

Tweets are also characterized by extensive use of abbreviations and slang specific to social-media (e.g. *ily* for 'I love you', *iono* for 'I don't know'). Such abbreviations may squash several parts-of-speech into one token, which poses a challenge to popular POS taggers.

In addition, due to the colloquial nature of user-created content, it is common that proper words are replaced by phonetically or morphologically similar ones (e.g.

'wtchng' instead of 'watching', 'gr8' instead of 'great'). Users may also use capitalization irregularities, deliberate spelling errors, punctuation irregularities and interjections as a means to express their sentiment, as in the following tweet:

Haha South Tampa is getting flooded hah-
WAIT A SECOND I LIVE IN SOUTH
TAMPA WHAT AM I GONNA DO WHAT AM
I GONNA DO FVCK #flooding

Lastly, tweets may contain a variety of tokens seen mainly in Twitter and other social media, such as: URLs; emoticons; Twitter hashtags, of the form #*tagname*, which the author may supply to label a tweet; Twitter at-mentions of the form @*user* which link to other Twitter users; and Twitter discourse functions such as RT ("re-tweet"), indicating that a tweet was originally posted by some other Twitter user, or ellipsis dots (...) at the end (or beginning) of a tweet, indicating that the tweet will be continued in a subsequent tweet by the same user. We note that hashtags and at-mentions can also serve as words or phrases within a tweet, as in:

Heard about the #earthquake, stay safe ev-
everyone.

Regarding URLs, all links posted in tweets are shortened using Twitter's link service, <http://t.co>, and are converted to a seemingly random 23 characters URL that redirects to the original web address.

2 Classification of Disaster-Related Tweets

In the first part of our work we developed a classifier that identifies *disaster-related* tweets, trained on a dataset of 10,877 labeled tweets. We used only tweets with label confidence ≥ 0.9 , which is about half of the original dataset. We experimented with Naive Bayes (NB), random forest (RF) and support vector machine (SVM) classifiers.

Naive Bayes (NB). A supervised probabilistic learning method that classifies according to *maximum a posteriori*. In this method we used *unigram* and *bigram* features.

Random Forest (RF). An ensemble learning method that uses some estimators (trees) for classification. In this method we used *unigram* and *bigram* features.

Support Vector Machine (SVM). A discriminative learning method that attempts to find the hyperplane that creates a maximal margin between positive and negative classifications on the training data, using penalty method for mistakes. In this method we used *unigram*, *bigram*, *tweets metadata* and *POS tagging* features.

2.1 Feature Extraction

To train the classifiers we extracted several features for each tweet:

- **Unigrams and bigrams** of tokens in tweet; we used a Python version of `Tokenizer`, tokenizer for Twitter data (Gimpel et al. [2], Myle Ott, 2013 [3]).
- **Tweet metadata**; hashtags, at-mentions and URLs; we parsed the tweet, crawled URLs and used the referred webpage title as supplementary information for our classifier.
- **Part-of-speech (POS) tags** (bigram and unigram); we used a twitter-specific tagset and tagger developed by Gimpel et al. [2].

Tokenization Splitting a tweet to tokens (separated ordered words) is hard due to irregular punctuation patterns and use of punctuation marks for emoticons. For example, the correct tokenization of 'hello (#hashtag)' is to the four tokens [hello , (, #hashtag ,)], but 'hello (: should be split only to the tuple [hello , (:].

`Tokenizer` is a tokenizer designed for Twitter text in English that addresses these issues. It was originally developed in Python by O'Connor et al., 2010 [1], then improved and ported to Java by Gimpel et al., 2011 [2] and later ported back to Python [3]. We use the last version by Myle Ott.

Metadata Extraction The hashtags, at-mentions, URLs and emoticons in a tweet carry information that may help better understand the subject and context. Therefore, for each tweet we created a vector of the following metadata features, which we found to be the most expressive:

- Does the tweet contain any links, and if so how many?
- Is 'https://twitter.com' one of the links (i.e. a link that refers to another tweet)?
- Does the tweet contain a user at-mention (@)?
- Does the tweet contain any hashtags (#), and if so how many?
- Does the tweet contain a happy emoticon (e.g. :D)?

In addition to these features, we extracted information from hashtags and URLs. We attempted to split hashtags to separate words looking for a CamelCase pattern (for example, #JeSuisCharlie → 'Je Suis Charlie') or words separated by underline. We note that this method is not exhaustive since Twitter users tend to create hashtags composed of joined words, all lower-case.

We also found that the domain name of the URLs in a tweet may give an indication to whether the tweet is disaster-related or not (for example, a link to

https://9gag.com is a negative hint). However, Twitter applies URL shortening on links, so for each shortened link in the dataset we attempted to reach the original Internet address. We also collected the HTML page title, which often contains the title of an article (for example, in news sites). We managed to expand 4,823 URLs out of 6,157 in the dataset.

For each tweet in the dataset we created an '*extended*' version where (1) hashtags are extracted, (2) URLs are expanded to original URL followed by the page title and (3) every user at-mention is replaced by the token `__USERREF__`. For example, the following is a tweet,

```
http://t.co/c1H7JECFrV @RoyalCaribbean
do your passengers know about the mass
murder that takes place in the #FaroeIs-
lands every year?
```

and its *extended version* is,

```
https://www.royalcaribbean.co.uk/ Holiday
Destinations - Cruise Destinations — Royal
Caribbean UK. __USERREF__ do your pas-
sengers know about the mass murder that
takes place in the Faroe Islands every
year?
```

Twitter POS Tagging We used a Twitter-oriented POS tagger written in Java (Gimpel et al., 2011 [2]). This tagger features a tagset that captures Twitter-specific properties. The tagset contains a set of 20 coarse-grained tags based on several treebanks, with some additional categories specific to Twitter, such as URLs and hashtags (the complete tagset is available in [2]). The tagger is a conditional random field (CRF), and its training involved manual annotation of tweets.

Unlike common English taggers, the Twitter tagger does not split contractions or possessives (which are much more common in social media than in newspaper corpora), hence the following tags are being used:

- **S** - nominal + possessive (e.g. books')
- **Z** - proper noun + possessive (e.g. America's)
- **L** - nominal + verbal (e.g. i know (= I don't know))
- **M** - proper noun + verbal (e.g. Mark'll)

In addition, the tag **G** is being used for social-media abbreviations, foreign words and Twitter 'garbage'.

The tagger exploits some phonetic and linguistic features that are useful when dealing with colloquial writing, such as:

- **Phonetic normalization** - Twitter includes many alternate spelling of words. The Metaphone algorithm was used to create a coarse phonetic normalization of words to simpler key.
- **Frequently-capitalized tokens** - Twitter users are inconsistent in their use of capitalization. Therefore, a lexicon of frequently capitalized tokens was

created to cluster different tokens that refer to the same word or entity.

2.2 Results

We divided the dataset into train (90%) and test (10%) sets. Each classification method was tested with *unigram* features and *bigram* features separately, and for the *SVM* classifier *tweets metadata* and *POS tagging* features were tested as well.

Accuracy is presented using 3 measures:

- **Total Accuracy** is the number of right classifications out of the total queries ($\frac{TP+TN}{TP+FP+TN+FN}$).
- **Positive Predictive Value (PPV)** is the number of right positive classifications out of total positive classifications ($\frac{TP}{TP+FP}$).
- **Negative Predictive Value (NPV)** is the number of right negative classifications out of total negative classifications ($\frac{TN}{TN+FN}$).

For both *Random Forest* and *SVM* classifiers we tuned the parameters using *grid search*. For *Random Forest* we tuned the number of estimators and for *SVM* we tuned the penalty constant.

Random Forest Results. Looking at the *Random Forest* results (fig. 1), we can easily see that the *Unigram* features have better accuracy and better *NPV*, while *Bigram* features have better *PPV* for any number of estimators. In addition, best results for total accuracy, *PPV* and *NPV* are achieved with 128, 64 and 128 estimators respectively.

SVM Results. Looking at the *SVM* results (fig. 2, fig. 3) we see that for *Unigram* features with or without *POS*, all measurements are becoming constant for $C \geq 10^4$ (penalty is too big to be used). In addition, best results for total accuracy, *PPV* and *NPV* are achieved with penalty constant with value 10^4 , 10^3 and 10^4 respectively. For *Bigram* features results are quite similar, except for a minor gap (around 1.5%) between using *POS* and not. Best results for total accuracy, *PPV* and *NPV* are exactly the same as in the *Unigram* features. We note that all classifiers used *tweets metadata* features as well.

Tables 1 and 2 present the best measurement result for each classifier; best total accuracy result (94.5%) is achieved with *SVM* classifier using *Bigram*, *tweets metadata* and *POS-tagging* features.

We accurately identified disaster-related tweets using various classifiers. It is worth mentioning that the extraction of *tweet metadata* (i.e. expansion of hashtags and URL crawling) and the use of *extended tweets* significantly improved the results; measurements on tweets before *extension* showed a classification accuracy around

82% - over 10% worse than after applying the pre-processing technique.

	Uni NB	Bi NB	Uni RF	Bi RF
accuracy	0.921	0.864	0.911	0.890
ppv	0.977	1.000	0.945	0.969
npv	0.891	0.813	0.892	0.856

Table 1: Disaster Classification Best Results: *Naive Bayes* and *Random Forest*

	Uni SVM	Uni+POS SVM	Bi SVM	Bi+POS SVM
accuracy	0.935	0.937	0.933	0.945
ppv	0.961	0.946	0.952	0.955
npv	0.935	0.938	0.926	0.939

Table 2: Disaster Classification Best Results: *SVM*

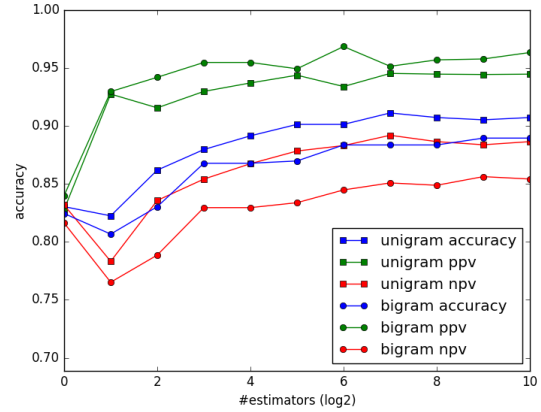


Figure 1: Disaster Classification - *Random Forest* (Unigram vs. Bigram)

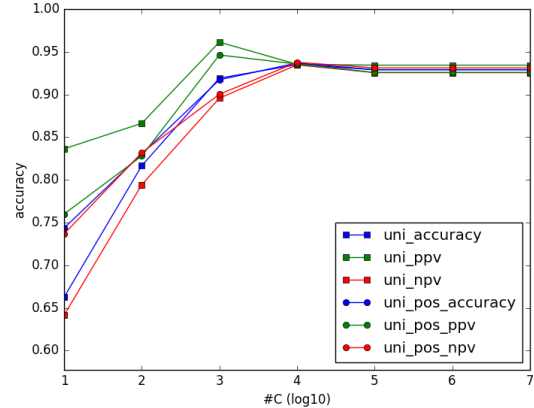


Figure 2: Disaster Classification - *SVM* (Unigram vs. Unigram and POS)

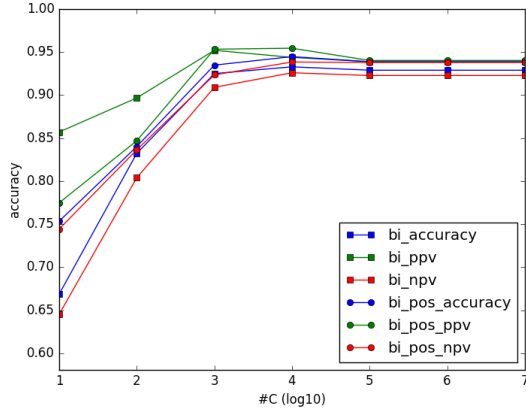


Figure 3: Disaster Classification - SVM (Bigram vs. Bigram and POS)

3 Sentiment Analysis of Tweets

Sentiment analysis tasks aim to determine the writer’s attitude, opinion or emotion, and are usually designed as a multi-class classification task (for example, among the labels “sad”, “happy”, “angry”, etc.). Another target may be classifying the *polarity* of a given text (for example, “positive” or “negative”).

We focused on the task of classifying disaster-related tweets to “objective” and “subjective” tweets. In large, we say that “objective” tweets contain mere facts, undeniable truths that are universally agreed upon, and that “subjective” tweets contain an opinion, sentiment or emotion towards some happening (e.g. sadness, worry, fear, sarcasm, etc.). For example, the following is an “objective” tweet:

FedEx no longer to transport bioterror germs in wake of anthrax lab mishaps
<http://t.co/P96rgBbaYL> #news #phone #apple #mobile

and the two following tweets are “subjective”:

@denisleary Not sure how these folks rush into burning buildings but I’m grateful they do. #TrueHeroes

@canagal Good to hear it’s back.. that storm’s been given you guys trouble though :(

The ability to differentiate between *objective* and *subjective* tweets may help in identifying individuals who witness some disastrous event or who are in need of help.

3.1 The Dataset

Our dataset of 10,877 tweets did not contain a label of *objective* / *subjective*, and we did not find any other satisfactory pre-labeled dataset of disaster-related tweets. Hence, we went on a mission to manually label all 2,100 disaster-related tweets from the dataset (that were labeled ‘disaster-related’ with confidence ≥ 0.9).

Each tweet was independently labeled by two participants; upon disagreement the third participant broke the tie. Around 80% of the tweets were labeled *objective* while only 20% were labeled *subjective*. This results from the large amount of tweets by news agencies in the dataset.

It is worth to mention that in some cases, subtleties in writing made it hard to decide whether a tweet is “objective” or “subjective”. For example, the following tweet is clearly “objective”,

Thunder lightening torrential rain and a power cut.

while this tweet is debatable,

Thunder lightening torrential rain and a power cut!

and the following is clearly “subjective”,

Thunder lightening torrential rain and a power cut! :(

3.2 Feature Extraction

We confirmed our hypothesis that the task of differentiating between “objective” and “subjective” disaster-related tweets can be achieved without examining the vocabulary of tweets, but merely by using POS tags and the following metadata features, which we extracted for each tweet:

- Presence and number of exclamation marks.
- Presence and number of question marks.
- Presence of URLs.
- Presence of emoticons (extracted using `emoticon.py`, Twitter NLP framework [4]).
- Number of digits.
- Number of capital words.
- Number of capital letters.
- Number of punctuation marks and symbols (!"#\$%&'()*+,-./:;=<?@[\\]^_`{|}~).
- Tweet’s length.
- POS tag statistics:
 - Number of adjectives.
 - Number of verbs.
 - Number of adverbs.
 - Number of pronouns.
 - Number of proper nouns.
 - Number of possessive endings.

3.3 Results

We trained *SVM* and *Random Forest* classifiers for this task, using $C = 10^4$ as a penalty constant for *SVM* and 128 estimators for *Random Forest*.

Once again the dataset was divided into train (90%) and test (10%) sets. Each classification method was tested with the previously mentioned features, using feature selection with *ANOVA F-test* model. We repeated the test several times, each time increasing the number of K best features selected by the feature selection model (out of 18 features total), and measuring accuracies (as mentioned in the previous *Identification* task). Our goal was to find which features are the most expressive for the task. We used the measures of total accuracy, *PPV* and *NPV*.

Random Forest. Looking at the *Random Forest* results (fig. 4), we can see that best total accuracy is achieved with 13 features, so for *NPV*, but best *PPV* is achieved with only 1 feature (might be due to the biased dataset). The most expressive features used are: number of exclamation marks, presence of exclamation mark, presence of question mark, presence of URL, presence of emoticon, number of digits, number of capital letters, number of punctuation marks and symbols, tweet’s length.

SVM. Looking at the *Random Forest* results (fig. 5), we can see that best total accuracy is achieved with 4 features only, so for *NPV*, but best *PPV* is achieved with only 1 feature as well (again, might be due to the biased dataset). The most expressive features used are: number of exclamation marks, presence of URL, presence of emoticon, number of punctuation marks and symbols.

Table 3 presents the best measurement result for each classifier. Best total accuracy result (87.1%) is achieved with *Random Forest* classifier using 13 features.

	RF (1)	RF (13)	SVM (1)	SVM (4)
accuracy	0.805	0.871	0.805	0.848
ppv	0.905	0.890	0.905	0.874
npv	0.500	0.750	0.500	0.667

Table 3: Objective/Subjective Classification Best Results (in parentheses - number of features used)

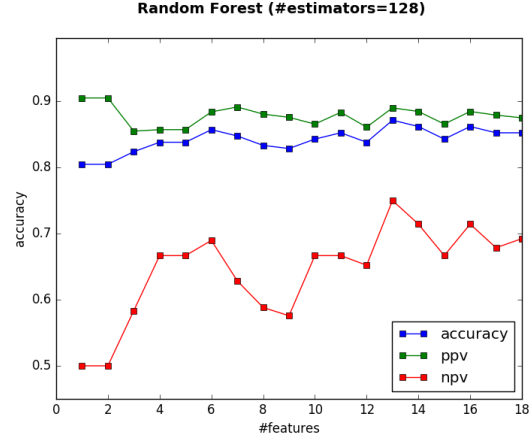


Figure 4: Objective/Subjective Classification - *Random Forest*

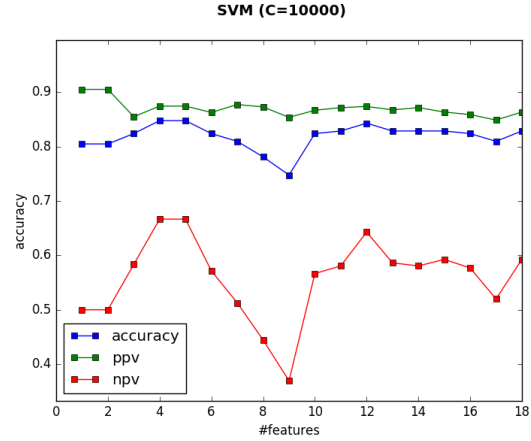


Figure 5: Objective/Subjective - *SVM*

4 Named-Entity Recognition in Tweets

We aim to extract relevant information from tweets we previously classified as disaster-related. Named-entity recognition (NER) allows us to extract names of people, organizations, locations and time expressions. Ritter et al., (*Named Entity Recognition in Tweets: An Experimental Study*, 2011 [5]) provide a method to extract named entities out of *tweet corpora*, by using a Twitter-specific POS tagset (similarly to the tagset we used in our previous classifiers). In this section, we present our attempt to use the Groningen Meaning Bank (GMB), which was trained on news and articles (rather than Twitter corpora), and extract named entities out of tweets.

Groningen Meaning Bank (GMB). In order to develop a named-entity recognizer we used the Groningen Meaning Bank² - a large corpus containing thousands of

texts in raw and tokenized formats, POS tags, named entities by lexical categories and much more. The top level categories of the corpus are:

- **geo** - Geographical Entity
- **org** - Organization
- **per** - Person
- **gpe** - Geopolitical Entity
- **tim** - Time indicator
- **art** - Artifact
- **eve** - Event
- **nat** - Natural Phenomenon

We used only the top-level categories, which are sufficient for extracting relevant information on disasters. As the database contains POS tags in NLTK format, we used NLTK to create POS tags for our tweets, rather than the Twitter tagger by Gimpel et al. [2].

Capitalization. A key feature for recognizing named entities capitalization. In case of tweets, which are written by non-professional writers, capitalization is much less reliable, and differs in style between twitter users. Capitalization irregularities in tweets can be the result of spelling errors, or may be used to express the writers sentiment or emotion.

To overcome this obstacle, we implemented an approach similar to that of Ritter et al., 2011 [5], and converted our data to all lower-case. Using capitalized words as features caused a great shift in the results - entities of different types where often grouped together, and non-entities where often identified as such. Manipulations on the capitalization feature extractor - such as announcing a word as capitalized, only if its surroundings are not all capitalized, performed poorly as well. By neutralizing this feature, and converting all the tweets to lower case letters, the classifier performance increased significantly.

4.1 Results

Four main categories of the GMB corpus are relevant for our purposes - *location (geo)*, *organization (org)*, *persons (per)* and *geopolitical entity (gpe)*. A fourth category, *time (tim)*, was somewhat ambiguous - as it described both time entities and irrelevant ones. Other categories did not manage to reach any level of internal agreement on any entity.

By using the GMB corpus as the training data, we performed named entity recognition on the previously used dataset of 10,877 labeled tweets. From those tweets, we singled out only those who were labeled as disaster-related. The main results were:

- *Geographical Entity* - California and Hiroshima were the dominant entities, as there were tweets about disasters in those places, followed by Japan, India, south, and Saudi Arabia.

- *Organization* - This category was composed of news companies, such as ABC and BBC, and by examples such as Washington, which was treated as an organization representing the U.S (rather than the city/state). Many user references are marked as organizations as well.
- *People* - The only pure result in this category is Obama.
- *Geopolitical Entity* - Composed mostly of user references, and nationalities - Israeli, Turkish and more.
- *Time Indicator* - ranged from a description of current events - wildfire or typhoon, to description of the 70th anniversary of Hiroshima atom bombing. Included some specific time references, such as Wednesday.

From those results, we can learn the location, nationality of involvement, associated organizations, and type of the disaster. We classified and manually checked the results in order to learn the false-alarm rate of our implementation, as presented in table 4. The most common entities are California, Hiroshima, ABC, Turkey, Northern, severe, refugio, Japan, Obama and ABC-News.

	geo	org	per	gpe	tim
Total	337	306	41	865	787
Success Rate	80%	57%	76%	49%	27%

Table 4: Named Entity Recognition Success Rate

5 Experimenting with Recent Tweets

In the last part of our work we tested our classifier that identifies *disaster-related* tweets and the NER classifier on real data that was recently posted on Twitter. We used Twitter’s Search API that allows developers to browse tweets that were posted during the previous 7 days, and filter them according to various categories such as language, location (by coordinates), keywords and more.

The Data. We created the following three tweet dataset:

- **Houston** - 1,501 tweets posted in the *Houston, TX* area during 29 Aug. - 3 Sept., containing at least one of the words [water, drown] and 500 random tweets
- **Miami** - 1,001 tweets posted in the *Miami, FL* area during 9 Sept., containing at least one of the words [evacuate, stay, safe, car] and additional 150 random tweets

For both datasets we avoided from ”re-tweeted” tweets (RT), to prevent repetitions of popular tweets.

From each of the datasets we used our used our classifier (see section 2) to take only *disaster-related* tweets. We checked the performance of our NER classifier by running it on the disaster-related tweets in each dataset, and presenting the most popular named-entities found. We ran both our NER classifier (trained on the GMB dataset), and the Twitter-oriented NER by Ritter et al. [5].

The most popular entities among the **Houston** tweets:

- **Our NER** - Houston, Mandatory, Twitter, Harvey, Water, Beaumont, Air Alliance, Info, Houston Water, Hurricane Harvey
- **Ritter et al.** - Houston, Twitter, Harvey, Hurricane Harvey, Google Maps, Beaumont, Houston Water, Katy, Air Alliance Houston, Texas

The most popular entities among the **Miami** tweets:

- **Our NER** - Florida, Irma, Miami, Twitter, Hurricane, Hurricane Irma, Leaving, Floridians arent, Cuba and Hurricaneirma
- **Ritter et al.** - Florida, Irma, Miami, Twitter, Hurricane Irma, Cuba, Florida Keys, tampa, Key West, Paramount Pictures Settles

The results show that both NER methods manage to identify the two recent large scale hurricanes that affected Houston (Hurricane Harvey) and Miami (Hurricane Harvey). The comparison starts to fail when comparing the uncommon named entities; this is not surprising, since we expected Ritter’s method, which is adjusted to Twitter corpora, to be superior. However, our method is sufficient for understanding the trivial details of large-scale disasters - location and type of disaster.

6 Conclusions

TODO: Any ideas of what should we write here?

The code of our project is available at <https://github.com/glrn/nlp-disaster-analysis>.

References

- [1] BRENDAN O’CONNOR, M. K., AND AHN, D. Tweetmotif: Exploratory search and topic summarization for twitter. In *Proc. ICWSM* (2010).
- [2] KEVIN GIMPEL, NATHAN SCHNEIDER, B. O. D. D. M. J. E. M. H. D. Y. J. F., AND SMITH, N. A. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proc. ACL* (2011).
- [3] OTT, M. Python port of Twokenizer. <https://github.com/myleott/ark-twokenize-py>, 2013.
- [4] RITTER, A. Twitter nlp tools. https://github.com/aritter/twitter_nlp, 2011.
- [5] RITTER, A., CLARK, S., MAUSAM, AND ETZIONI, O. Named entity recognition in tweets: An experimental study. In *Proc. EMNLP* (2011).

Notes

¹“Disasters on social media” Dataset by CrowdFlower: Contributors looked at over 10,000 tweets culled with a variety of searches like “ablaze”, “quarantine”, and “pandemonium”, then noted whether the tweet referred to a disaster. <https://www.crowdflower.com/wp-content/uploads/2016/03/socialmedia-disaster-tweets-DFE.csv>

²“Groningen Meaning Bank” Dataset by University of Groningen: comprises of thousands of texts in raw and tokenized formats, tags for part of speech, named entities and lexical categories, and discourse representation structures compatible with first-order logic. <http://gmb.let.rug.nl/>