Comparison of Word Embeddings and Sentence Encodings as Generalized Representations for Crisis Tweet Classification Tasks

Hongmin Li*
Department of Computer Science
Kansas State University
hongminli@ksu.edu

Xukun Li
Department of Computer Science
Kansas State University
xukun@ksu.edu

Doina Caragea
Department of Computer Science
Kansas State University
dcaragea@ksu.edu

Cornelia Caragea
Department of Computer Science
Kansas State University
ccaragea@ksu.edu

ABSTRACT

Many machine learning and natural language processing approaches, including supervised and domain adaptation algorithms, have been proposed and studied in the context of filtering crisis tweets. However, the application of these approaches in practice is still challenging due to the time-critical requirements of emergency response operations, and also to the diversity and unique characteristics of emergency events. To address this limitation, we explore the idea of building "generalized" classifiers for filtering crisis tweets, classifiers which can be pre-trained and ready to use in real-time, while they generalize well on tweets from future disasters. We propose to achieve this objective using a simple feature-based adaptation approach, where tweets are represented as dense numeric vectors of reduced dimensionality using either word embeddings or sentence encodings. Given that several types of word embeddings and sentence encodings exist, we compare tweet representations corresponding to different word embeddings and sentence encodings with the goal of understanding what embeddings/encodings are more suitable for use in crisis tweet classification tasks. Our experimental results on three crisis tweet classification tasks suggest that the tweet representations based on GloVe embeddings produce better results than the representations that use other embeddings, when employed with traditional supervised learning algorithms. Furthermore, the GloVe embeddings trained on crisis data produce better results on more specific crisis tweet classification tasks (e.g., tweets informative versus non-informative), while the GloVe embeddings pre-trained on a large collection of general tweets produce better results on more general classification tasks (tweets relevant or not relevant to a crisis).

Keywords

Word Embeddings, Sentence Encodings, Reduced Tweet Representation, Crisis Tweet Classification

INTRODUCTION

The value of social media during crisis situations has been well established among researchers and emergency response practitioners (Castillo 2016; Homeland Security 2014; Imran et al. 2015; Reuter et al. 2018). Social media represents an important crisis communication tool (Palen and Hughes 2018), and also a golden mine of critical first-hand information for situational awareness. However, given the fast accumulated user generated data, and easy spread of rumors and misinformation on popular platforms such as Twitter and Facebook, information

*corresponding author

overload is a big technical challenge preventing extensive adoption of social media in practice (Plotnick et al. 2015; Reuter et al. 2015; Tapia and Moore 2014). Furthermore, the time-critical requirements of crisis response make the identification and processing of situational awareness information in real-time even more challenging.

To reduce the information overload, we need automatic tools to identify social media information that is relevant to disasters, informative, reliable, and actionable, and to categorize such information into specific situational awareness categories. Significant progress has been made in this direction by employing machine learning and natural language processing (NLP) techniques, such as traditional supervised machine learning (Caragea et al. 2014; Imran et al. 2013; Starbird et al. 2010; Verma et al. 2011), deep learning (Nguyen et al. 2016a,b), and domain adaptation (Imran et al. 2016; Li et al. 2017b, 2015). Systems and other crisis-related resources have been developed as well, e.g. (Terpstra et al. 2012), AIDR[1] and EU COMRADES project[2].

However, there are still challenges in applying such methods in real-time, especially during the early hours of emergencies (Reuter et al. 2018). First, supervised machine learning algorithms, especially the data hungry deep learning methods, often rely on large amounts of in-domain labeled data, which is usually human annotated and hardly available in the early hours of a crisis. Second, models trained on past disasters may not generalize well to an emerging target disaster (Imran et al. 2015). While domain adaptation approaches represent better alternatives as compared to supervised learning, they usually have more hyper-parameters that need to be tuned, and some guidelines need to be provided to enable their application in real-time in crisis situations (Li et al. 2017a).

A very attractive solution to these problems consists of ready-to-use “generalized" crisis classifiers that can be used directly on future disasters, with high accuracy. In this paper, we study a simple domain adaptation approach based on pre-trained word embeddings and existing sentence-level encoding which are used to produce a dense, reduced representation of crisis tweets classifications. In this context, “simple” means that the domain adaptation between prior crisis data (training, or “source") and current crisis data (test, or “target") is based just on generalized feature representations obtained either through simple averaging of the embeddings of the words in the tweet, or through the use of pre-trained sentence embedding models on the whole tweet. As the vocabulary of a target crisis is somewhat different from the vocabularies of prior crises, the traditional bag-of-words representation obtained on a prior crisis does not include all the keywords that appear in the target crisis, and can result in poor performance.

Vector representations through word embeddings or sentence-level encodings can capture the semantic meaning of the text, and therefore can be used to address the limitation of the bag-of-words representation. When using embeddings to represent vocabulary words, words with similar semantic are close in the word embeddings vector space. Thus, similar texts from the source and target domains (i.e., texts which use different words but have similar semantic meaning to human), can be captured through similar word embeddings. Word embeddings that are pre-trained on large news or Wikipedia corpora are available, and can be used directly to represent tweets from a new emerging crisis. It is also possible to train the word embeddings specifically on a corpus of crisis tweets. Furthermore, more sophisticated pre-trained models that produce sentence-level encodings have also shown good performance on various NLP tasks (Conneau et al. 2017). We can directly use such sentence-level encodings to get reduced representations for crisis tweet classification tasks. Such adaptation framework, based on word embeddings or sentence encodings, is not only extensible to future crises, but also flexible, in the sense that after the reduced tweet representations are obtained, supervised or other domain adaptation approaches, such as (Mazloom et al. 2018), can be applied on top of the reduced representations.

In this paper, we focus on two classification tasks, which represent essential first filtering steps in the process of extracting useful information from social media data posted during a crisis, specifically, the task of classifying crisis tweets as relevant versus irrelevant, and the task of classifying crisis tweets as informative versus non-informative. We experiment with three commonly used types of word embeddings, both pre-trained and trained on our own crisis tweet corpus, along with different average/maximum/minimum approaches for aggregating the word embeddings. We also experiment with pre-trained sentence-level encoding models, to directly obtain vector representations for crisis tweets. We use the resulting representations to train supervised tweet classification models, and evaluate the models in a realistic setting, where the training of the model is done on tweets from prior crises, and the testing of the model is done on a new crisis.

We experiment with different types of word embeddings and sentence encodings, and different types of supervised classifiers, with the goal of gaining insights the embeddings that are most suitable to use with crisis data and also into the supervised models that result in better “generalized” classifiers. Experimental results suggest that, in general, GloVe word embeddings, either pre-trained on Twitter data or trained on our crisis tweet corpus, work better with the simple methods for aggregating word embeddings. Furthermore, we observe that the tweet representations

[1]http://aidr.qcri.org/
[2]http://comrades-project.eu

based on sentence encodings can significantly improve the accuracy of some classifiers (e.g., Gaussian Naive Bayes) as compared with the representations based on simple word embedding aggregations, suggesting great potential for sentence encodings. To our knowledge, research in this direction is still limited, especially in the area of crisis informatics. Our study is the first to apply recent state-of-art pre-trained sentence-level encoding models to crisis tweet classification, and also the first one that evaluates three different types of word embeddings (most of the existing works use just one type of word embedding, e.g., Word2Vec). Thus, our work can be seen as an important step towards building "generalized" crisis classifiers that are ready-to-use in a new crisis situation.

To summarize, our main contributions are as follows:

- We explore the idea of "generalized" classifiers for real-time filtering of crisis tweets, and propose a simple feature-based adaptation framework for building generalized classifiers using word or sentence embeddings.
- We study different representations of crisis tweets, using different word and sentence embeddings, as well as different supervised learning algorithms on top of the representations, as described in Section 2 (Methods). In addition to using pre-trained word embeddings, we also train crisis specific word embeddings with our own crisis tweet corpus. The resulting embeddings can be used for other crisis tweet classification tasks.
- We perform extensive experiments with the representations generated from word and sentence embeddings to understand what representations are more suitable for crisis tweet classification tasks. We also experiment with different supervised learning algorithms to gain insights into classifiers that generalize well from embeddings. The datasets used in the experiments are described in Section 3 (Datasets), the experimental setup is described in Section 4 (Experimental Setup) and finally the experimental results are presented and discussed in Section 5 (Experimental Results and Discussion).

METHODS

Similar to the deep learning concepts, the ideas of distributed word representations are not new, but they have become more popular given the more powerful computing resources available nowadays and the accumulation of large datasets in many application domains.

Tweet Representations Using Word Embeddings

Currently, research on word embeddings is still one of the most popular topics in the NLP area. Here, we adopt three types of word embeddings widely used in the NLP community:

1. *Word2Vec* (Mikolov et al. 2013a,b);
2. *GloVe* (Pennington et al. 2014);
3. *FastText* (Bojanowski et al. 2017; Mikolov et al. 2018).

With each type, we use both existing embeddings pre-trained on Wikipedia or Twitter data, and also crisis word embeddings trained specifically on our crisis tweet corpus. Subsequently, we use several simple ways to combine the word embeddings into reduced tweet representations, as described below:

a) *Mean*: We average the embeddings of each word in the tweet along each dimension. Thus, a tweet vector will have the same dimension as a word vector/embedding.

b) *MinMaxMean* (MMM): In addition to mean, we also take the minimum and maximum over all the words in a tweet, along each dimension of the word vectors. Each aggregation, min/max/mean, will produce a vector that has the same dimension as the word vectors. We concatenate the vectors corresponding to min/max/average, respectively, and obtain a tweet vector whose dimension is three times the dimension of the word vector.

c) *Tf-idf-Mean*: We assign tf-idf (term frequency - inverse document frequency) weights to the words in a tweet, and calculate the weighted average of the word embeddings along each dimension (where the contribution of a word is proportional to its tf-idf weight).

Tweet Representations Using Sentence Embeddings

Given that tweets resemble sentences as they are relatively short, we also experiment with recent approaches on sentence-level embeddings/encodings. Specifically, we use the following models to get tweet representations:

1. *Smooth Inverse Frequency* (SIF) (Arora et al. 2017): The representation produced by this approach can be seen as the weighted average of the word vectors, modified by removing the projections of the average vectors on their first principal component ("common component removal') (Arora et al. 2017). This simple method has been shown to beat more sophisticated models, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), on semantic textual similarity tasks.

2. *InferSent* (Conneau et al. 2017): This approach consists of universal sentence representation models trained with natural language inference data, using different network architectures such as Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRU), or Bi-directional LSTM networks (BiLSTM). We use the pre-trained model published by the authors to generate tweet representations with this approach.

3. *Universal Sentence Encoder on TensorFlow* (tfSentEncoder) (Cer et al. 2018): The universal sentence encoding models are trained with the same data as the InferSent models, but with different model architectures.

The reader is referred to the original articles, which introduced the above-mentioned sentence encoding approaches, as such details are beyond the scope of this article. But in general, the purpose and usage of the universal sentence encoders are similar to purpose/usage of the word embeddings, as the pre-train universal sentence encoders can help with NLP and text classification tasks that rely on sentences (Cer et al. 2018).

Tweet Classification Using Supervised Learning

Finally, once we obtain the vector representations for tweets, any supervised learning algorithm (or even more complex domain adaptation algorithms) can be used to learn classification models using the tweet reduced representations. Here, for the sake of simplicity, and also to satisfy the real-time prediction requirements, we choose to experiment with four traditional supervised machine learning algorithms:

1. Gaussian Naive Bayes (GNB);

2. Random Forest (RF);

3. K Nearest Neighbors (KNN);

4. Support Vector Machines (SVM).

We included the Naive Bayes algorithm in our study as it doesn't require hyper-parameter tuning, and can be trained in linear time. The other algorithms, Random Forest, KNN, and especially SVM, all have been extensively used in text classification tasks. More complex models based on deep learning will be used in future work.

DATASETS

We used three datasets in our study, specifically: 1) CrisisLexT6 (Olteanu et al. 2014); 2) CrisisLexT26 (Olteanu et al. 2015); and 3) 2CTweets (Schulz et al. 2017). CrisisLexT6 and CrisisLexT26 are available from the CrisisLex project website[3]. We obtained the 2CTweets dataset directly from the authors.

The first dataset, CrisisLexT6, is a collection of English tweets collected during six disasters that occurred between October 2012 and July 2013 in USA, Canada and Australia, as shown in the first part of Table 1. Tweets were collected using the Twitter Streaming API based on disaster-related keywords and geo-locations of the affected areas. There are approximately 10,000 tweets for each disaster, all manually labeled as *on-topic* (i.e., relevant) or *off-topic* (i.e., irrelevant) using the crowdsourcing platform CrowdFlower (currently, renamed FigureEight).

The second dataset, CrisisLexT26, is a collection of tweets posted during 26 crisis events that happened in 2012 or 2013, with most events having between 2,000 and 4,000 tweets. These tweets were also collected using filtering keywords, and labeled by CrowdFlower workers according to informativeness (i.e., *informative* or *non-informative*), information types (e.g., *caution and advice*, *infrastructure damage*), and information sources (e.g., *Governments*,

[3]http://crisislex.org/data-collections.html

Table 1. Statistics about the datasets (CrisisLexT6, CrisisLexT26, and 2CTweets), before and after cleaning

	Before Cleaning			After Cleaning		
CrisisLexT6	On-topic	Off-topic	Total	On-topic	Off-topic	Total
2012_Sandy_Hurricane	6138	3870	10008	5443	3757	9200
2013_Queensland_Floods	5414	4619	10033	3324	4530	7854
2013_Boston_Bombings	5648	4364	10012	4824	4301	9125
2013_West_Texas_Explosion	5246	4760	10006	4123	4711	8843
2013_Oklahoma_Tornado	4827	5165	9992	4101	5111	9212
2013_Alberta_Floods	5189	4842	10031	4550	4745	9295
CrisisLexT26	Informative	Non-Inf.	Total	Informative	Non-Inf.	Total
2012_Colorado_wildfires	685	268	953	665	252	917
2013_Queensland_floods	728	191	919	681	183	864
2013_Boston_bombings	417	512	929	397	489	886
2013_West_Texas_explosion	472	439	911	444	390	834
2013_Alberta_floods	685	298	983	665	284	949
2013_Colorado_floods	768	157	925	736	147	883
2013_NY_train_crash	904	95	999	684	88	772
2CTweets	Yes	No	Total	Yes	No	Total
Memphis	361	721	1082	333	699	1032
Seattle	800	1404	2204	739	1293	2032
NYC	413	1446	1859	373	1411	1784
Chicago	214	1270	1484	202	1254	1456
San Francisco	304	1176	1480	290	1146	1436
Boston	604	2216	2820	586	2123	2709
Brisbane	689	1898	2587	667	1746	2413
Dublin	199	2616	2815	189	2574	2763
London	552	2444	2996	490	2287	2777
Sydney	852	1991	2843	832	1947	2779

NGOs). As we used English word embeddings, we only selected 7 events from the 26 events in our study, as shown in the middle part of Table 1, and only focused on the task of classifying tweets as *informative* or *non-informative*.

The third dataset, 2CTweets, is a collection of tweets about incidents, such as car crash, fire or shooting, which happened in 10 different cities, as shown in the last part of Table 1. Tweets were labeled as incident related (*Yes*) or not (*No*). Given that incidents in different cities most likely involve local named entities, such as local street names, adaptation is needed to enable generalization of classifiers between different cities (Schulz et al. 2017).

To benefit the most from pre-trained embeddings, ideally, one should preprocess the data to be embedded the same way as the corpus that was used for training the embeddings. However, for the pre-trained Word2Vec and FastText embeddings, the original preprocessing performed is not well documented, and we used directly the raw tweets to obtain the corresponding word embeddings representations. As opposed to that, the preprocessing script used when training GloVe word embeddings on Twitter data is available from the GloVe's website[4]. Thus, we applied the same preprocessing for all our datasets when using pre-trained GloVE or crisis-specific word embeddings. Specifically, we used a Python version of the GloVe's Ruby preprocessing script, which consists of the following main steps: 1) replacing URLs and user mentions with placeholders <url> and <user>, respectively; 2) replacing different emoticons with placeholders such as <smile>, <lolface>, <sadface>, <neutralface> and <heart>, respectively; 3) replacing numbers with a placehoder <number>; 4) changing uppercase words to lowercase words, and tagging them with the tag <allcaps>, for example, "HURRICANE" becomes "hurricane <allcaps>"; 5) similarly, tagging punctuation repetitions, elongated words and hashtags, for example "!!!" becomes "! <repeat>", "soooo" becomes "so <elong>" and "#HurricaneSandy" becomes "<hashtags> hurricanesandy"; 6) converting all tweets to lowercase and tokenizing them based on whitespace using the Stanford Tokenizer. Finally, duplicate tweets identified after preprocessing were removed. The statistics of each class in each event dataset, before and after the preprocessing, together with the total number of tweets in the dataset, are shown in Tables 1 for the three datasets, respectively.

[4]https://nlp.stanford.edu/projects/glove/preprocess-twitter.rb

Our crisis tweet corpus for training crisis specific word embeddings are also processed the same way. Besides, the three datasets described above, the corpus also includes tweets that we collected through the Twitter Streaming API during several disasters that happened in Fall 2017, specifically Hurricane Harvey, Hurricane Irma, Hurricane Maria, and also Mexico Earthquake. The total corpus contains approximately 5.8 million tweets. Most of these tweets are crisis related, but there is also some inherent noise due to the fact that the streaming was keyword-based.

EXPERIMENTAL SETUP

For each of the three datasets, we experimented with three types of word embeddings, both existing pre-trained embeddings, and custom embeddings trained on crisis tweets, as well as three sentence embedding models, for a total of 9 different representations for the tweets. The notations and details of the embeddings used are as follows:

- ***Word2Vec***: denotes the pre-trained Word2Vec embeddings. Specifically, we used the set of embeddings trained on Google news. The dimension of the Word2Vec embeddings is 300.
- ***CrisisW2V***: denotes the Word2Vec embeddings trained with our crisis tweet corpus. We used the Gensim implementation of Word2Vec, and trained a CBOW (Continuous Bag Of Words) model, also with dimension 300. We used the default values for all the other Gensim parameters.
- ***GloVe***: denotes the GloVe embeddings pre-trained on Twitter data. We also experimented with GloVe embeddings pre-trained on Wikipedia data, but the results were worse than those obtained with the Twitter embeddings, and will not be shown. Furthermore, there are four different sets of pre-trained GloVe Twitter embeddings, corresponding to four dimensions: 25, 50, 100 and 200. Our preliminary results showed that 50 or 100 dimensions give similar results, generally better or comparable with the results obtained with 25 or 200 dimensions. We only show the results obtained with embeddings with 100 dimensions.
- ***CrisisGloVe***: denotes GloVe embeddings trained on our crisis tweet corpus. We use the GloVe original package to learn the embeddings, minimum word frequency count is 10, maximum iterations is set to 100 and window size is 10. We only show the results obtained with the 100 dimensional embeddings with same reason as ***GloVe*** setting.
- ***FastText***: denotes the pre-trained FastText embeddings. We experiment with the 1 million word vectors trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset. The dimension is 300.
- ***CrisisFastText***: denotes the embeddings trained on the crisis tweet corpus with the FastText original package. We used CBOW as in CrisisW2V, with dimension 300, and default values for all the other parameters.
- ***SIF***: denotes the SIF approach, which is considered to be a baseline for sentence embeddings. The original paper used GloVe embeddings pre-trained on the Common Crawl data. However, we used GloVe embeddings with 100 dimensions, pre-trained on Twitter data, for consistency with the ***GloVe*** setting.
- ***InferSent***: denotes the universal sentence representation model trained with natural language inference data. We used the pre-trained model (Conneau et al. 2017), which encodes tweets into 4096 dimensional vectors.
- ***tfSent***: denotes the universal sentence encoder from (Cer et al. 2018). We used the encoder available from TensorFlow TF-hub, which encodes tweets into 512 dimensional vectors.

As discussed in Section 3 (Dataset section), all experiments are running on processed tweets except experiments of ***Word2Vec*** and ***FastText*** which are on original raw tweets corresponding to the cleaned tweets due to preprocessing steps are not explicit for these two type pre-trained embeddings[5]. For each type of word embedding, we further used the three aggregation approaches mentioned in the previous section, specifically, Mean, Min/Max/Mean (MMM) and Tf-idf-Mean (Tfifd) to convert the word embeddings into reduced tweet representations. Furthermore, for each word-based or sentence-based representation, we experimented with four supervised learning algorithms from the scikit-learn library: 1) Gaussian Naive Bayes (GNB); 2) Random Forest (RF), where the number of trees (n_estimators) was set to 100 (and default values were used for the other parameters); 3) K Nearest Neighbors (KNN), for which the default number of neighbors was 5; 4) Support Vector Machine (SVM), with default parameters, including cost parameter $C = 1$ and RBF kernel.

Leave-one-out: We focused on three classification tasks, corresponding to the three datasets in our study: CrisisLexT6, CrisisLexT26, and 2CTweets. We evaluated different tweet representations based on word embeddings

[5]We also run these experiments on the cleaned tweets but the results are slight better or worse depending on the datasets, and therefore not showed here.

or sentence embeddings, on each of the three datasets. A leave-one-out setting was used for evaluation to simulate a real scenario. Namely, for each dataset, in a particular experiment, we selected one event as the target test data, and used the rest of the events from that dataset for training. For example, when Hurricane Sandy from CrisisLexT6 was selected as test, the other 5 disasters from CrisisLexT6 were used for training. Each disaster was left out in one experiment, therefore, the number of experiments conducted for one dataset is given by the number of events in the dataset. The results reported for a dataset are averaged over all experiments conducted on that dataset. Intuitively, the averages should provide an indication of how well the models built generalize to future events.

EXPERIMENTAL RESULTS AND DISCUSSION

Given the variety of word embeddings that are available in the machine learning and NLP communities, we aim to understand how the performance of traditional classifiers vary with different types of word embeddings, and whether re-training the embeddings on crisis data is necessary or not. Furthermore, given that text classification tasks (e.g., sentiment analysis) have been shown to benefit from sentence-level embeddings pre-trained on data from other NLP tasks, we also aim to understand how different sentence-level embeddings perform when used for crisis tweet classification tasks. Finally, towards the ultimate goal of building "generalized" classifiers, we aim to gain insights into how different classifiers handle different types of embeddings, and what algorithms best handle the embeddings, in general. The observations that we seek to draw from our experiments could be used to provide guidelines for future works that need to choose among different types of embeddings and supervised classifiers, as well as guidelines for the adoption of the framework for domain adaptation through embeddings in practice.

The results of the experiments that used tweet representations based on word embeddings are shown in Table 2, and the results based on sentence embeddings are shown in Table 3. We analyzed the results of the experiments presented in Tables 2 and 3 driven by several research questions, as described below.

- *Among the pre-trained word embeddings (i.e., Word2Vec, GloVe and FastText), which embeddings and aggregations work better, in general, with different supervised classifiers?*

  By analyzing the results in Table 2 by column, we can compare the performance of different types of embeddings and aggregation methods (i.e., Mean, MinMaxMean or Tfidf weighted averaging), when used with different algorithms. For each dataset and each classifier, we underscore the best value obtained with that classifier on the dataset to observe which types of embeddings/aggregations perform well for specific classifiers and across different classifiers. The more underscored values in a column, the better the corresponding embedding performs. From Table 2, we can see that the GloVe embeddings work better for two datasets out of three. Specifically, for CrisisLexT6, GloVe pre-trained Twitter embeddings work the best, while for CrisisLexT26 dataset, CrisisGloVe embeddings are better. However, for the 2CTweets dataset, the Word2Vec embeddings (either pre-trained or trained on crisis data) are better across several supervised classifiers, although the performance of the GloVe embeddings is very close or even better in some cases (e.g., with Random Forests). One possible explanation for this may be that the 2CTweets dataset is mostly about incidents that appear as local news in a city, as opposed to large-scale disasters. Intuitively, the GloVe embeddings pre-trained on Twitter may capture well disasters such as Hurricane Sandy (given that tweets about disasters may be part of the training set), but may not capture well the local incidents.

  Regarding the different approaches to aggregate word embeddings, the performances vary with the aggregation approach, and it cannot be claimed that one approach is better than the others, in general. However, if we focus on the best types of word embeddings for each dataset (CrisisLexT6, GloVe column; CrisisLexT26, CrisisGloVe column; and 2CTweets, Word2Vec, CrisisW2V columns), it can be observed that the MinMaxMean (MMM) is generally better than or very close to the other two aggregation approaches, although sometimes the best values are achieved by Mean aggregation way.

- *Given the existing pre-trained embeddings, do crisis tweet classification tasks benefit from embeddings trained specifically on a crisis tweet corpus?*

  When comparing the pre-trained embeddings with the embeddings trained on the crisis tweet corpus, we can see that for CrisisLexT6 the embeddings pre-trained on a general corpus are better, while for CrisisLexT26 and 2CTweets the embeddings trained on the crisis tweet corpus are better. In particular, for CrisisLexT26, the results with CrisisW2V and CrisisFastText are better for almost all classifiers, as compared to the results obtained with the corresponding Word2Vec and FastText pre-trained embeddings. While the CrisisGloVe embeddings are not always better than the pre-trained Twitter GloVe embeddings, they can still achieve competitive performance when used with Random Forest or SVM classifiers.

Table 2. Average accuracy values (and standard deviation) for three datasets, CrisisLexT6, CrisisLexT26, and 2CTweets, using the leave-one-out evaluation strategy, with six types of word embeddings and four supervised classifiers. For each dataset, the best value of each column is highlighted in bold font. Furthermore, underscored values represent the best values among those corresponding to a particular classifier. The more underscored values in a column, the better the corresponding word embedding performs as compared to other embeddings. Also, the more bold values one type classifier has for a dataset, the better that type of classifier is for that dataset.

CrisisLexT6		Word2Vec	CrisisW2V	GloVe	CrisisGloVe	FastText	CrisisFastText
GNB	Mean	0.808±0.079	0.757±0.073	0.834±0.076	0.783±0.085	0.825±0.072	0.764±0.059
	MMM	0.797±0.062	0.798±0.058	0.828±0.067	0.791±0.065	0.798±0.060	0.774±0.060
	Tfidf	0.687±0.047	0.753±0.088	0.831±0.059	0.785±0.103	0.662±0.046	0.752±0.065
RF	Mean	0.841±0.087	0.833±0.092	0.875±0.074	**0.835±0.107**	0.847±0.093	0.815±0.085
	MMM	0.824±0.091	0.832±0.088	0.859±0.096	0.816±0.116	0.808±0.095	0.819±0.074
	Tfidf	0.824±0.087	0.821±0.101	0.872±0.071	0.827±0.121	0.824±0.094	0.805±0.097
KNN	Mean	0.847±0.041	0.820±0.060	0.838±0.034	0.825±0.082	0.823±0.035	0.779±0.048
	MMM	0.863±0.058	0.821±0.067	0.848±0.059	0.829±0.099	0.850±0.059	0.746±0.043
	Tfidf	0.801±0.037	0.801±0.051	0.819±0.035	0.800±0.069	0.791±0.034	0.751±0.044
SVM	Mean	0.846±0.077	**0.863±0.102**	**0.894±0.047**	0.826±0.125	0.829±0.081	**0.847±0.110**
	MMM	**0.873±0.066**	0.858±0.095	0.890±0.059	0.833±0.131	0.862±0.078	0.837±0.083
	Tfidf	0.872±0.057	0.795±0.112	0.888±0.043	0.828±0.124	**0.875±0.054**	0.792±0.104
CrisisLexT26		Word2Vec	CrisisW2V	GloVe	CrisisGloVe	FastText	CrisisFastText
GNB	Mean	0.807±0.058	0.796±0.060	0.821±0.061	0.817±0.066	0.798±0.062	0.785±0.058
	MMM	0.787±0.067	0.796±0.062	0.841±0.046	0.832±0.048	0.793±0.060	0.773±0.051
	Tfidf	0.806±0.056	0.814±0.053	0.812±0.053	0.811±0.066	0.808±0.057	0.815±0.050
RF	Mean	0.836±0.045	0.849±0.046	0.850±0.046	0.858±0.040	0.843±0.047	0.841±0.048
	MMM	0.838±0.046	0.853±0.043	0.856±0.045	0.861±0.046	**0.844±0.050**	0.843±0.042
	Tfidf	0.824±0.046	0.850±0.042	0.830±0.046	0.845±0.047	0.816±0.046	0.842±0.040
KNN	Mean	0.823±0.060	0.830±0.050	0.836±0.056	0.834±0.055	0.813±0.070	0.826±0.044
	MMM	0.805±0.063	0.800±0.039	0.838±0.058	0.822±0.057	0.809±0.065	0.769±0.062
	Tfidf	0.797±0.061	0.830±0.048	0.830±0.052	0.821±0.067	0.796±0.074	0.845±0.044
SVM	Mean	0.703±0.148	**0.862±0.039**	0.857±0.045	**0.863±0.042**	0.702±0.148	**0.855±0.042**
	MMM	0.797±0.082	0.857±0.038	**0.861±0.046**	0.861±0.041	0.712±0.143	0.841±0.048
	Tfidf	**0.844±0.046**	0.838±0.045	0.841±0.063	0.841±0.060	0.839±0.043	0.838±0.048
2CTweets		Word2Vec	CrisisW2V	GloVe	CrisisGloVe	FastText	CrisisFastText
GNB	Mean	0.881±0.046	0.785±0.058	0.850±0.053	0.829±0.053	0.849±0.047	0.737±0.056
	MMM	0.877±0.051	0.801±0.057	0.844±0.056	0.828±0.061	0.838±0.053	0.744±0.059
	Tfidf	0.825±0.054	0.818±0.048	0.852±0.053	0.854±0.050	0.786±0.055	0.791±0.054
RF	Mean	0.896±0.042	0.886±0.039	0.890±0.037	0.891±0.033	0.886±0.043	0.877±0.045
	MMM	**0.905±0.036**	0.905±0.036	**0.906±0.033**	0.903±0.032	**0.905±0.038**	0.883±0.045
	Tfidf	0.864±0.056	0.885±0.040	0.884±0.043	0.876±0.046	0.857±0.059	0.876±0.044
KNN	Mean	0.901±0.032	0.891±0.036	0.880±0.039	0.887±0.043	0.879±0.041	0.869±0.046
	MMM	0.900±0.033	0.882±0.041	0.871±0.049	0.885±0.042	0.881±0.028	0.822±0.047
	Tfidf	0.884±0.032	0.881±0.041	0.875±0.040	0.876±0.044	0.870±0.044	0.866±0.044
SVM	Mean	0.859±0.045	0.907±0.036	0.893±0.036	0.898±0.034	0.777±0.082	**0.912±0.031**
	MMM	0.890±0.037	**0.921±0.025**	0.904±0.037	0.902±0.036	0.840±0.052	0.892±0.040
	Tfidf	0.903±0.038	0.889±0.047	0.900±0.039	**0.906±0.034**	0.892±0.041	0.876±0.060

One possible reason that the GloVe crisis-specific embeddings perform better than the pre-trained embeddings on CrisisLexT26 , but not on the CrisisLexT6 dataset, may be the fact that the CrisisLexT6 classification task (relevant to disasters or not relevant) is more general, and thus its vocabulary may be better covered by the general Twitter corpus used for the pre-trained embeddings. As opposed to that, the CrisisLexT26 tasks are more specific to crises and benefit more from crisis embeddings.

- *Among the sentence encodings (i.e., SIF, InferSent, tfSent), which encodings work better, in general, with different classifiers? How do the sentence encodings compare with the word embeddings?*

  By comparing columns in Table 3, we can see that the InferSent sentence encoder is better for CrisisLexT26 and 2CTweets, while the simple SIF encoder is generally better for the CrisisLexT6 dataset. InferSent generates sentence encodings with 4096 dimensions, a significantly larger number of dimensions as compared to the SIF sentence encoder (which uses just 100 dimensions), and the tfSent encoder (which uses 512 dimensions). Based on our prior experimentation with CrisisLexT6 and CrisisLexT26, a relatively small number of features is needed to discriminate between relevant and non-relevant tweets in CrisisLexT6, while a larger number of features is needed to discriminate between crisis informative and non-informative tweets in CrisisLexT26. Our prior observations match with the current study which suggests that a sentence encoder that produces vectors with a small number of dimensions (SIF) is useful for CrisisLexT6, while an encoder that produces vectors with a large number of dimensions (InferSent) is useful for CrisisLexT26. Thus, in general, the choice of the sentence encoder may be related to the choice of the number of dimensions used by the encoder, which in turn depends on difficulty of the classification task at hand.

  To evaluate word embeddings versus sentence encodings, we compared Tables 2 and 3, and observed that the best values for a dataset are generally obtained using word embeddings. This result is counter-intuitive, as one would expect the sentence-level encodings to better capture the content of a tweet. The reason for this result may be related to the fact that we do not perform hyper-parameter tuning for classifiers such as RF and SVM. For CrisisLexT6, the Gaussian Naive Bayes (GNB) classifier, which does not have any hyper-parameters, the tfSent model produces an average accuracy of 88%, which is a 5% improvement of the best accuracy obtained with word embeddings. Thus, sentence encoders have great potential for crisis tweet classification tasks, but more experimentation and hyper-parameter tuning is needed to better evaluate their benefits.

- *Among the classifiers studied, GNB, RF, KNN and SVM, which one benefits more from embeddings?*

  When using word embeddings, the SVM classifier performs the best overall, regardless of the type of embedding and aggregation employed. The results of the RF classifier are sometimes the best for a dataset, or very close to the results of SVM. When using sentence encodings, both RF and SVM classifiers work well for the CrisisLexT6 dataset, while for the other two datasets, the RF classifier works better. However, we believe that hyper-parameter tuning might improve the results of SVM, making this classifier more competitive also when using sentence encodings. Gaussian Naive Bayes doesn't work well with simple word embedding aggregations, but performs well with sentence encodings on the larger CrisisLexT6 dataset. The results of the KNN classifier are better than the results of Gaussian Naive Bayes, but worse than those of RF and SVM, in general. The less-competitive performance of KNN may be due to the differences between the events in a dataset. Given that we are using several disasters to train a model, and then we test the model on a left-out test disaster, the different disasters used in training may bring in noise with respect to the test disaster. Even though the word embeddings or sentence encodings are meant to bridge the semantic gap between tweets, KNN is still sensitive to noise as it is making its classifications based on the nearest neighbors selected. If the nearest neighbors are noisy, the classification can be wrong. Thus, overall, our study suggests that SVM or RF are good choices as traditional supervised learning classifiers, but hyper-parameter tuning may be needed to achieve best performance.

RELATED WORK

In NLP, word embeddings are used either to represent the features for a standard traditional classifier, or as initializations in a deep neural network, which will subsequently tune the initial values through backpropagation. Existing NLP works that compare different word embeddings aim to evaluate the quality of the embeddings across different NLP tasks (Baroni et al. 2014; Nayak et al. 2016; Schnabel et al. 2015). Evaluation approaches can be grouped into two categories: intrinsic evaluation and extrinsic evaluation (Schnabel et al. 2015). Intrinsic evaluations measure the quality of word embeddings by directly computing the correlation between semantically and geometrically related terms, usually through pre-collected inventories of query terms (Schnabel et al. 2015). For instance, Word2Vec is evaluated on a word similarity task in the original paper which introduced it. Baroni et al. (2014) and Faruqui and Dyer (2014) are also focusing on intrinsic evaluations using a variety of query inventories.

Table 3. Average accuracy values (and standard deviation) for three datasets, CrisisLexT6, CrisisLexT26, and 2CTweets, using the leave-one-out evaluation strategy, with three types of sentence embeddings and four supervised classifiers. For each dataset, the best value of each column is highlighted in bold font. Furthermore, underscored values represent the best values among those corresponding to a particular classifier. The more underscored values in a column, the better the corresponding word embedding performs as compared to other embeddings. Also, the more bold values one type classifier has for a dataset, the better that type of classifier is for that dataset.

		SIF	InferSent	tfSent
CrisisLexT6	GNB	0.834±0.065	0.806±0.060	0.881±0.061
	RF	0.885±0.057	**0.883±0.048**	0.879±0.065
	KNN	0.856±0.035	0.873±0.048	0.877±0.052
	SVM	**0.896±0.042**	0.870±0.061	**0.893±0.050**
CrisisLexT26	GNB	0.800±0.020	0.821±0.049	0.800±0.053
	RF	**0.813±0.070**	**0.853±0.044**	**0.847±0.041**
	KNN	0.810±0.060	0.822±0.052	0.818±0.071
	SVM	0.810±0.085	0.702±0.148	0.702±0.148
2CTweets	GNB	0.835±0.032	0.859±0.050	0.880±0.050
	RF	**0.881±0.053**	0.898±0.039	**0.895±0.046**
	KNN	0.876±0.030	**0.901±0.036**	0.887±0.036
	SVM	0.880±0.046	0.849±0.055	0.864±0.070

In extrinsic evaluations, word embeddings are used as input features to a downstream task, and the evaluation of the word embeddings is done according to the performance metrics specific to that task (Schnabel et al. 2015). For instance, GloVe embeddings are evaluated on part-of-speech tagging and named-entity recognition tasks (Pennington et al. 2014). Nayak et al. (2016) also proposed to evaluate word embeddings using a standardized suite of characteristic downstream tasks, so that the evaluation is more likely to generalize to real-world applications of the embeddings. Together, a thorough intrinsic evaluation of word vectors, and a limited extrinsic evaluation (Schnabel et al. 2015) showed that the performance on two downstream tasks (noun phrase chunking and sentiment classification) is not consistent, and may not be consistent with intrinsic evaluations either. The authors suggested that training specific embeddings to optimize a specific objective is generally better for downstream tasks. This is also one reason we compare pre-trained embeddings with crisis-specific embeddings on crisis tweet classification. Schnabel et al. (2015) showed, in the context of sentiment classification of movie reviews, that CBOW (a Word2Vec model) is better than GloVe and some other distributed word representations. In their study, the word embeddings for sentiment classification were used to generate embedding-only features for each movie review by computing a linear combination of word embeddings weighted by the number of times that word appeared in the review.

Other prior studies that used word embeddings on tweet classification tasks also generated the vector representations of tweets by averaging the word embedding vectors along each dimension for all the words in a tweet (Boom et al. 2016; Wang et al. 2015; Yang et al. 2016). The average representation was usually compared with the weighted average word embedding representation, and/or with approaches that use the word embeddings in deep learning models such as Convolutional Neural Networks (CNN). For example, Boom et al. (2016) focused on learning short text representations by averaging each dimension of the word embeddings. They also experimented with averaged embeddings concatenated with minimum and/or maximum aggregated embeddings, and proposed a weight-based approach that performs better on their semantically related and non-related pairs of words from Twitter data. Yang et al. (2016) studied the effect of the configuration used to train and generate word embeddings on a Twitter election classification task, where average word embedding representations of tweets, used with the SVM classifier, were compared with CNN models. The results suggested that the CNN models outperformed the SVM models.

While many previous studies have focused on text representations through the means of word embeddings or sentence encoders, the research on the usability of word/sentence embeddings for crisis tweet representation and classification is limited. Nguyen et al. (2016) used Word2Vec embeddings to initialize CNN models trained to classify crisis tweets. They also used crisis-specific Word2Vec embeddings trained on a corpus with approximately 60,000 tweets, and showed that the crisis-specific embeddings are slightly better than the pre-trained embeddings. While we also compared pre-trained embeddings with crisis-specific embeddings, we experimented and compared a bigger variety of word embeddings used subsequently with traditional supervised classifiers. Furthermore, our training corpus for crisis-specific embeddings was much larger than the one used by Nguyen et al. (2016). Finally,

to the best of our knowledge, we are the first to use recent advances in sentence-level encoding models (Cer et al. 2018; Conneau et al. 2017) in the context of crisis tweet representation and classification.

CONCLUSIONS AND FUTURE WORK

In this paper, we study the feasibility of building "generalized" crisis classifiers through the means of word embeddings and sentence encodings. Towards this goal, we compared three types of word embeddings (Word2Vec, GloVe and FastText) and three approaches to aggregate the word embeddings into tweet representations (specifically, Mean, MinMaxMean, and TfidfMean). We performed an extrinsic evaluation, where the embeddings were used with four traditional machine learning algorithms (Gaussian Naive Bayes, Random Forest, K-Nearest Neighbors and Support Vector Machines) to learn generalized classifiers for crisis tweet classification tasks. We used both word embeddings pre-trained on corpora such as Google News, Wikipedia or Twitter, and crisis-specific embeddings trained on a large tweet corpus. We observed that the crisis-specific embeddings are more suitable for more specific crisis-related tasks, while the pre-trained embeddings are more suitable for more general classification tasks, where not all the tweets classified are crisis related. Among the three types of word embeddings (Word2Vec, GloVe and FastText), the GloVe embeddings performed the best overall on the three datasets used in our study (CrisisLexT6, CrisisLexT26 and 2CTweets). Specifically, the pre-trained GloVe embeddings worked better for the more general classification task in CrisisLexT6, while the CrisisGloVe performed better for the more specific classification tasks in CrisisLexT26. Furthermore, the SVM classifier was shown to make best use of the embeddings in terms of generalizing to data from future disasters. In addition to word embeddings, we also experimented with recent models for sentence encoding, and showed that the sentence encoders have great potential for being used in crisis tweet classification tasks, although they may require more extensive hyper-parameter tuning. In particular, the sentence encodings worked well with the GNB classifiers which do not require hyper-parameter tuning, and can be potentially used in real-time in a crisis situation.

To conclude, our study can be used to provide insights into how the proposed "generalized" classifiers can be used in a real crisis situation. For an emergent disaster, the first classification task to be addressed is the classification of tweets as relevant or non-relevant to the disaster (similar to CrisisLexT6). Subsequently, relevant tweets can be classified as informative and non-informative (CrisisLexT26), and further into specific situational awareness categories such as injured people, damaged infrastructure, etc. Our preliminary results suggest that the relevant versus non-relevant classification task can be addressed using a "generalized" classifier that employs Twitter pre-trained GloVe embeddings with MinMaxMean tweet representations provided to the SVM classifier. Furthermore, the informative versus non-informative classification task can be addredded with a "generalized" classifier that employs CrisisGloVe embeddings, also with MinMaxMean tweet representations provided to the SVM classifier.

Classifiers for categorizing tweets in specific situational awareness categories will be designed in future work. Furthermore, hyper-parameter tuning will be performed for all tasks to improve the performance of the "generalized" classifiers based on word embeddings, and especially the performance of the classifiers based on sentence encodings. In addition to traditional learning algorithms, we will also experiment with deep neural networks, such as Convolutional Neural Networks and Recurrent Neural Networks, in future work. Finally, given our goal of building "generalized" classifiers, we plan to include a bigger variety of crisis events in the training dataset of the classifiers.

ACKNOWLEDGEMENTS

The computing for this project was performed on the Beocat Research Cluster at Kansas State University and Amazon Web Services (AWS). We thank the National Science Foundation for support from grant CNS-1429316, which funded the cluster. We also thank the National Science Foundation and Amazon Web Services for support from grants IIS-1741345 and IIS-1802284, which supported the research and the computation in this study. Finally, we thank the National Science Foundation for support from the grants IIS-1526542 and CMMI-1541155, which also supported this research. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either express or implied, of the National Science Foundation or AWS. We also wish to thank our anonymous reviewers for their constructive comments.

REFERENCES

Arora, S., Liang, Y., and Ma, T. (2017). "A Simple but Tough-to-Beat Baseline for Sentence Embeddings". In:

Baroni, M., Dinu, G., and Kruszewski, G. (2014). "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 238–247.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146.

Boom, C. D., Canneyt, S. V., Demeester, T., and Dhoedt, B. (2016). "Representation learning for very short texts using weighted word embedding aggregation". In: *CoRR* abs/1607.00570. arXiv: 1607.00570.

Caragea, C., Squicciarini, A. C., Stehle, S., Neppalli, K., and Tapia, A. H. (2014). "Mapping moods: Geo-mapped sentiment analysis during hurricane sandy". In: *11th Proceedings of the International Conference on Information Systems for Crisis Response and Management, University Park, Pennsylvania, USA, May 18-21, 2014.*

Castillo, C. (2016). *Big Crisis Data: Social Media in Disasters and Time-Critical Situations*. Cambridge University Press.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). "Universal Sentence Encoder". In: *CoRR* abs/1803.11175. arXiv: 1803.11175.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In: *CoRR* abs/1705.02364. arXiv: 1705.02364.

Faruqui, M. and Dyer, C. (2014). "Community evaluation and exchange of word vectors at wordvectors. org". In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 19–24.

Homeland Security (2014). *Using Social Media for Enhanced Situational Awareness and Decision Support*. Virtual Social Media Working Group and DHS First Responders Group.

Imran, M., Castillo, C., Diaz, F., and Vieweg, S. (2015). "Processing Social Media Messages in Mass Emergency: A Survey". In: *ACM Comput. Surv.* 47.4, 67:1–67:38.

Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., and Meier, P. (2013). "Practical extraction of disaster-relevant information from social media". In: *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, pp. 1021–1024.

Imran, M., Mitra, P., and Srivastava, J. (2016). "Cross-Language Domain Adaptation for Classifying Crisis-Related Short Messages". In: *13th Proceedings of the International Conference on Information Systems for Crisis Response and Management, Rio de Janeiro, Brasil, May 22-25, 2016.*

Li, H., Caragea, D., and Caragea, C. (2017a). "Towards Practical Usage of a Domain Adaptation Algorithm in the Early Hours of a Disaster". In: *14th Proceedings of the International Conference on Information Systems for Crisis Response and Management, Albi, France, May 2017.*

Li, H., Caragea, D., Caragea, C., and Herndon, N. (2017b). "Disaster Response Aided by Tweet Classification with a Domain Adaptation Approach". In: *Journal of Contingencies and Crisis Management* Special Issue on HCI in Critical Systems.In press.

Li, H., Guevara, N., Herndon, N., Caragea, D., Neppalli, K., Caragea, C., Squicciarini, A. C., and Tapia, A. H. (2015). "Twitter Mining for Disaster Response: A Domain Adaptation Approach". In: *12th Proceedings of the International Conference on Information Systems for Crisis Response and Management, Krystiansand, Norway, May 24-27, 2015.*

Mazloom, R., Li, H., Caragea, D., Imran, M., and Caragea, C. (2018). "Classification of Twitter Disaster Data Using a Hybrid Feature-Instance Adaptation Approach". In:

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). "Efficient Estimation of Word Representations in Vector Space". In: *CoRR* abs/1301.3781. arXiv: 1301.3781.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2018). "Advances in Pre-Training Distributed Word Representations". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* Pp. 3111–3119.

Nayak, N., Angeli, G., and Manning, C. D. (2016). "Evaluating word embeddings using a representative suite of practical tasks". In: *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pp. 19–23.

Nguyen, D. T., Al-Mannai, K., Joty, S. R., Sajjad, H., Imran, M., and Mitra, P. (2016a). "Rapid Classification of Crisis-Related Data on Social Networks using Convolutional Neural Networks". In: *CoRR* abs/1608.03902.

Nguyen, D. T., Joty, S. R., Imran, M., Sajjad, H., and Mitra, P. (2016b). "Applications of Online Deep Learning for Crisis Response Using Social Media Information". In: *CoRR* abs/1610.01030.

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 Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014*.

Olteanu, A., Vieweg, S., and Castillo, C. (2015). "What to Expect When the Unexpected Happens: Social Media Communications Across Crises". In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. CSCW '15. Vancouver, BC, Canada: ACM, pp. 994–1009.

Palen, L. and Hughes, A. L. (2018). "Social Media in Disaster Communication". In: *Handbook of Disaster Research*. Ed. by H. Rodríguez, W. Donner, and J. E. Trainor. Cham: Springer International Publishing, pp. 497–518.

Pennington, J., Socher, R., and Manning, C. D. (2014). "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Plotnick, L., Hiltz, S. R., Kushma, J. A., and Tapia, A. H. (2015). "Red Tape: Attitudes and Issues Related to Use of Social Media by U.S. County-Level Emergency Managers". In: *12th Proceedings of the International Conference on Information Systems for Crisis Response and Management, Krystiansand, Norway, May 24-27, 2015*.

Reuter, C., Hughes, A. L., and Kaufhold, M.-A. (2018). "Social Media in Crisis Management: An Evaluation and Analysis of Crisis Informatics Research". In: *International Journal of Human–Computer Interaction* 34.4, pp. 280–294. eprint: https://doi.org/10.1080/10447318.2018.1427832.

Reuter, C., Ludwig, T., Friberg, T., Pratzler-Wanczura, S., and Gizikis, A. (2015). "Social Media and Emergency Services?: Interview Study on Current and Potential Use in 7 European Countries". In: *IJISCRAM* 7.2, pp. 36–58.

Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). "Evaluation methods for unsupervised word embeddings". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 298–307.

Schulz, A., Guckelsberger, C., and Janssen, F. (2017). "Semantic Abstraction for generalization of tweet classification: An evaluation of incident-related tweets". In: *Semantic Web* 8.3, pp. 353–372.

Starbird, K., Palen, L., Hughes, A. L., and Vieweg, S. (2010). "Chatter on the red: what hazards threat reveals about the social life of microblogged information". In: *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW 2010, Savannah, Georgia, USA, February 6-10, 2010*, pp. 241–250.

Tapia, A. H. and Moore, K. (2014). "Good Enough is Good Enough: Overcoming Disaster Response Organizations' Slow Social Media Data Adoption". In: *Computer Supported Cooperative Work (CSCW)* 23.4, pp. 483–512.

Terpstra, T., Vries, A. de, and R. Stronkman, G. P. (2012). "Towards a realtime Twitter analysis during crises for operational crisis management". In: *Proceedings of the 9th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2012), Vancouver, Canada*, pp. 1–9.

Verma, S., Vieweg, S., Corvey, W. J., Palen, L., Martin, J. H., Palmer, M., Schram, A., and Anderson, K. M. (2011). "Natural Language Processing to the Rescue? Extracting "Situational Awareness" Tweets During Mass Emergency". In: *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*.

Wang, P., Xu, J., Xu, B., Liu, C., Zhang, H., Wang, F., and Hao, H. (2015). "Semantic Clustering and Convolutional Neural Network for Short Text Categorization". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*. The Association for Computer Linguistics, pp. 352–357.

Yang, X., MacDonald, C., and Ounis, I. (2016). "Using Word Embeddings in Twitter Election Classification". In: *CoRR* abs/1606.07006. arXiv: 1606.07006.